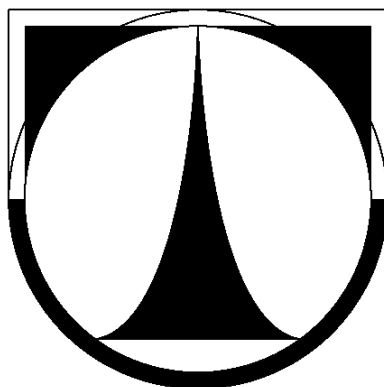


TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií



Vývoj univerzálního rozhraní pro LabVIEW

Bakalářská práce

Michal Jadrný

Liberec

2012

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Vývoj univerzálního rozhraní pro LabVIEW

Development of Universal Interface for LabVIEW

Bakalářská práce

Autor: **Michal Jadrný**

Vedoucí práce: Ing. Lenka Kretschmerová Ph.D.

V Liberci dne 18. května 2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal Jadrný**
Osobní číslo: **M09000059**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronické informační a řídicí systémy**
Název tématu: **Vývoj univerzálního rozhraní pro LabView**
Zadávací katedra: **Ústav řízení systémů a spolehlivosti**

Z á s a d y p r o v y p r a c o v á n í :


1. Proved'te rešerši možnosti využití různých rozhraní pro přenos dat pro LabView.
2. Proved'te rešerši možnosti přístupu k LabView programům přes webové rozhraní
3. Navrhněte univerzální rozhraní pro LabView, které bude přístupné přes webové rozhraní s definicí přístupových práv přihlášeného uživatele (user, admin atd.).
4. Navrhněte postup aktivace modulů zařízení (ze stanoveného úložiště) při jejich připojení k PC.
5. Navrhněte ovládání aktivních modulů zařízení v závislosti na přístupových právech přihlášeného uživatele.

Rozsah grafických prací: **dle potřeby**
Rozsah pracovní zprávy: **cca 40 stran**
Forma zpracování bakalářské práce: **tištěná/elektronická**
Seznam odborné literatury:


1. **Havlíček, J., Vlach, J., Vlach, M., Vlachová, V.: Začínáme s LabVIEW , BEN - technická literatura, ISBN 978-80-7300-245-9**
2. **Kainka, B.: USB - měření, řízení a regulace pomocí sběrnice USB. BEN - technická literatura, ISBN 80-7300-073-3**
3. **Technická dokumentace LabView - webovská rozhraní**

Vedoucí bakalářské práce: **Ing. Lenka Kretschmerová, Ph.D.**
Ústav řízení systémů a spolehlivosti
Konzultant bakalářské práce: **Jan Hřebíček**
Fyzikální ústav Akademie věd

Datum zadání bakalářské práce: **14. října 2011**
Termín odevzdání bakalářské práce: **18. května 2012**


prof. Ing. Václav Kopecký, OSc.
děkan




doc. Ing. Libor Tůma, CSc.
vedoucí ústavu

V Liberci dne 14. října 2011

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení, apod.).

Jsem si vědom toho, že užít své bakalářské práce, či poskytnout licenci k jejímu využití, mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím a konzultantem bakalářské práce.

V Liberci dne 18. května 2012

Podpis:

Poděkování

Na tomto místě bych rád poděkoval Ing. Lence Kretschmerové, Ph.D. za vedení a odbornou pomoc při vzniku bakalářského práce, za trpělivost a nespočetné množství rad a nápadů, kterými mě po celý školní rok zásobovala. Poděkování patří také konzultantu práce panu Janu Hřebíčkoví z Fyzikálního ústavu Akademie věd České republiky a panu Doc. Ing. Miroslavu Svobodovi, CSc. z Ústavu řízení systémů a spolehlivosti Fakulty mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci za jeho přínosnou pomoc během úvodního seznámení s programováním v prostředí LabVIEW.

Velké díky patří i celé mojí rodině za každodenní trpělivost, vstřícnost a obrovskou podporu nejenom při studiu na Technické univerzitě v Liberci.

Abstrakt

Bakalářská práce řeší návrh a realizaci univerzálního rozhraní v prostředí LabVIEW. V první části práce se čtenáři seznámí s nejpoužívanějšími rozhraními pro přenos dat, jejichž podporu má LabVIEW implementovanou, druhá kapitola pojednává o možnostech vzdáleného přístupu a řízení aplikace, což je jedním ze stěžejních bodů zadání práce. Další části práce se zabývají samotným vývojem modulárního programu s možností vzdáleného řízení. Je definováno několik skupin uživatelů s odlišnými přístupovými právy. Jednotlivé funkční části programu jsou popsány a doplněny návody na jejich úpravu. Součástí bakalářské práce jsou i autonomní moduly pro ovládání laboratorních měřicích přístrojů, na kterých je demonstrována funkčnost celého programu a možnost využití různých rozhraní pro přenos dat.

Klíčová slova

LabVIEW, měřicí rozhraní, modulární systém, ovládání měřicích přístrojů, sběrnice, vzdálené řízení

Abstact

Bachelor work describes design and realization of a universal interface in LabVIEW. In the first part, readers familiar with the most widely used interfaces for data transfer, whose support is implemented in LabVIEW, the second chapter deals with possibilities of remote access and application control, which is one of the main points of work assignment. The rest of the work deals with the real development of a modular program which supports remote control. In the program are defined several groups of users with different access rights. Each functional part of the program are described and supplemented by instructions for their preparation. The bachelor work also includes modules for autonomous control of laboratory instrumentation, which demonstrates the functionality of the program and the possibility to use different interfaces for data transfer.

Key words

LabVIEW, measuring interface, modular system, measuring instrument control, bus, remote control

Obsah

Prohlášení.....	- 4 -
Poděkování.....	- 5 -
Abstrakt.....	- 6 -
Abstact	- 7 -
Obsah	- 8 -
Úvod.....	- 9 -
1. Programovací a vývojové prostředí LabVIEW	- 10 -
1.1. Základní popis uživatelského rozhraní.....	- 10 -
2. Rozhraní pro přenos dat pro LabVIEW	- 13 -
2.1. Sériové komunikační rozhraní COM	- 13 -
2.2. USB	- 15 -
2.3. Rozhraní GPIB	- 18 -
2.4. Speciální měřicí karty.....	- 20 -
3. Možnosti přístupu k LabVIEW programů přes webové rozhraní	- 21 -
3.1. Připojení ke vzdálené ploše.....	- 21 -
3.2. Aplikace TeamViewer.....	- 22 -
3.3. LabVIEW Remote Panel	- 24 -
4. Návrh univerzálního rozhraní	- 28 -
4.1. Systém řízení přístupu	- 28 -
4.2. Přístupové heslo	- 30 -
4.3. Rozcestník modulů.....	- 31 -
4.4. Předávání vybraných hodnot.....	- 32 -
4.5. Logování do souboru.....	- 33 -
5. Aktivace modulů.....	- 35 -
6. Moduly pro měřicí přístroje.....	- 37 -
6.1. Generátor signálu Agilent 33220A	- 37 -
6.2. Generátor signálu Tektronix AFG3102.....	- 39 -
6.3. Multimetr Agilent 34410A.....	- 40 -
6.4. Laboratorní zdroj Agilent E3631A	- 41 -
7. Ovládání modulů v závislosti na přístupových právech.....	- 42 -
Shrnutí.....	- 46 -
Závěr	- 47 -
Seznam použité literatury	- 48 -
Seznam obrázků.....	- 50 -
Seznam tabulek.....	- 52 -

Úvod

Vzdálené řízení a ovládání prvků se v technických oborech stává již každodenní realitou. S tím se do popředí dostává potřeba různých druhů přístupů k takto řešeným procesům, které každý z nás běžně používá např. při společném využívání PC několika uživateli.

Cílem této bakalářské práce je naprogramovat univerzální (měřicí) rozhraní v prostředí LabVIEW podle specifikací Fyzikálního ústavu Akademie věd České republiky, který je zadavatelem práce. Od rozhraní se očekává, že bude umožňovat vzdálený přístup a řízení celého programu stejně tak, jako při fyzickém ovládání programu. V programu bude dále definováno několik uživatelských účtů s odlišnými přístupovými právy. Účty budou zabezpečeny heslem a díky nim bude možné, aby konkrétní uživatel ovládal pouze ty části programu, které úzce souvisejí s jeho prací a vyhnul se tak zásahům do programu, které mu nepřísluší. Program bude ve své podstatě grafická nadstavba nad moduly, jeho budoucí funkčnost bude dána použitými moduly. Jednotlivé moduly se aktivují po připojení daného zařízení k řídicímu PC. Aktivní moduly budou zobrazeny v hlavní nabídce programu, odkud je bude možné spouštět.

Programování tvoří majoritní část práce, ale nejdříve bych se rád zaměřil na krátký popis prostředí LabVIEW a práce v něm, který bude potřebný pro pozdější pochopení programu, a popsal některá rozhraní, používaná pro přenos dat. Jedná se o nejběžnější a nejrozšířenější rozhraní typu USB, sériová linka a rozhraní GPIB, které se využívá v měřicí technice. Dalším důvodem k výběru těchto rozhraní je fakt, že LabVIEW má pro ně přímo implementovanou podporu, díky čemuž se zjednoduší práce se sběrem dat.

Mimo rozhraní pro přenos dat se budu také věnovat způsobům vzdáleného přístupu k LabVIEW. Jednak programu, který je součástí operačního systému Windows, dále zástupci aplikací třetích stran a samozřejmě funkci, která je připravena přímo v LabVIEW.

V dalších částech práce se budu věnovat samotnému programování, popisu úskalí, na která jsem během práce narazil a jejich řešení. Nebude chybět podrobný popis jednotlivých funkcí programu, části zdrojového kódu i ukázky uživatelského rozhraní. Jako součást práce vzniknou i moduly pro ovládání měřicích přístrojů, aby bylo možné otestovat a demonstrovat možnosti programu.

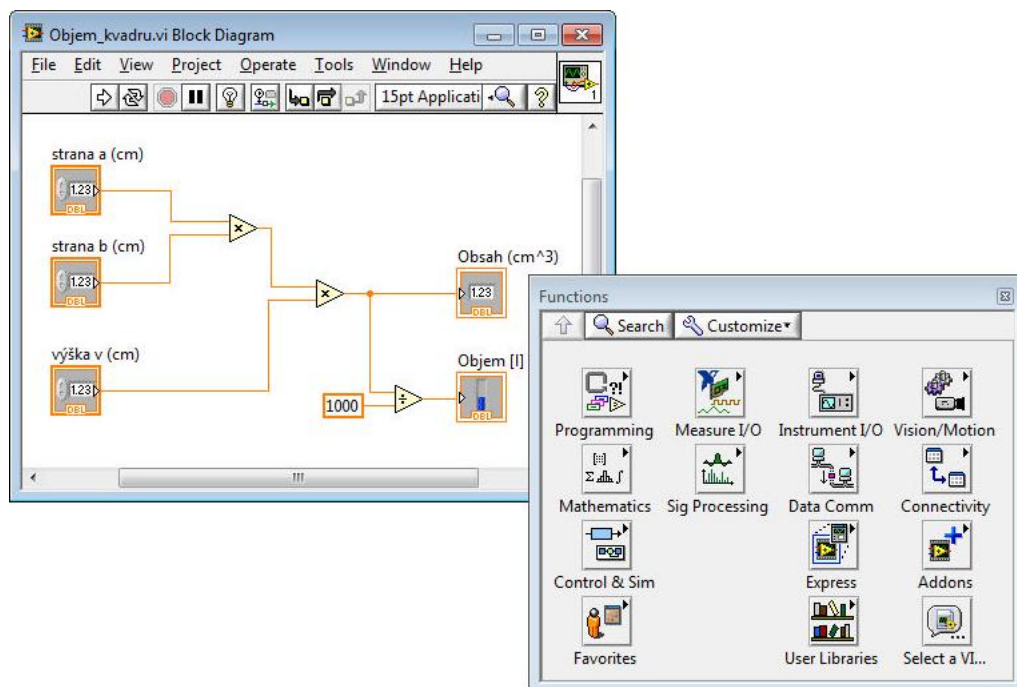
1. Programovací a vývojové prostředí LabVIEW

LabVIEW (z anglického Laboratory Virtual Instruments Engineering Workbench; česky: laboratorní pracoviště virtuálních přístrojů) je programovací a vývojové prostředí od americké firmy National Instruments, které bylo poprvé představeno v roce 1986. Jedná se o grafické vývojové prostředí, někdy označováno jako G-jazyk („grafický jazyk“). Impulzem k vývoji tohoto prostředí byla snaha přiblížit technikům programování tak, že pokud jsou schopni zapsat své požadavky do blokového diagramu, mohou podobným způsobem zapsat i celý program.

Vznikl tedy programovací nástroj, který umožňuje, podobně jako blokový diagram, vytvořit program pouhým propojováním předdefinovaných bloků. Po více než dvaceti pěti letech vývoje se z LabVIEW stal jeden z nejrozšířenějších průmyslových nástrojů, který slouží k řízení a vizualizaci různě složitých technologických procesů, k měření a analýze signálů, navíc je vhodný i pro programování složitých systémů. Zajímavostí je, že firma NI vyvinula, ve spolupráci s firmou LEGO, speciální verzi programu sloužící k programování mobilního robota *LEGO Mindstorm NTX*.

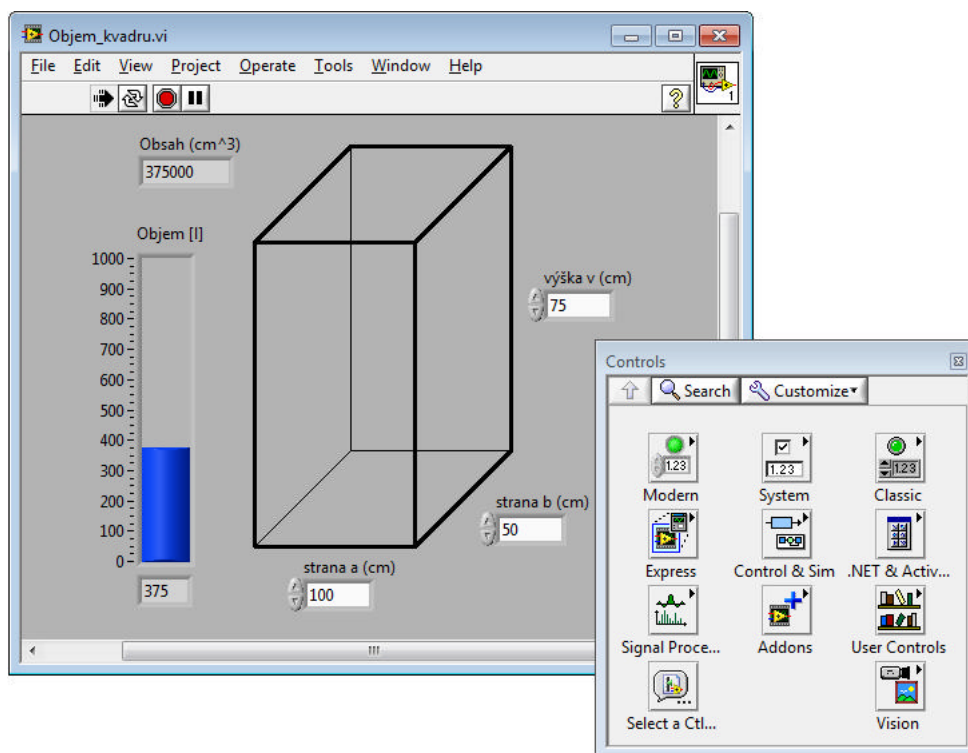
1.1. Základní popis uživatelského rozhraní

Programovací rozhraní je rozděleno na dvě části – *Block Diagram* (blokový diagram) a *Front Panel* (čelní panel). V blokovém diagramu probíhá veškeré programování, oproti jiným programovacím jazykům s rozdílem, že programovací příkazy jsou v LabVIEW nahrazeny grafickými symboly. Dalším výrazným rozdílem oproti běžnému textovému programování jsou pravidla, podle kterých program běží. U textového programování jsou příkazy prováděny v tom pořadí, v jakém jsou uvedeny v programu – sekvenční programování. Při programování v LabVIEW se příslušná akce bloku provede v okamžiku, kdy jsou na všech jeho vstupech dostupná data. Pokud blokové schéma obsahuje několik oddělených částí, můžeme se setkat s tím, že se v jeden okamžik provádí více nezávislých výpočtů. Tato vlastnost LabVIEW umožňuje rychlejší běh programu ve srovnání se sekvenčním programováním. V případě potřeby i LabVIEW umožňuje běh programu po sekvencích. Toho využijeme v případě, že potřebujeme různé části programu provádět vždy v určitém sledu bez ohledu na připravenost vstupních dat.



Obr. 1: Okno Block Diagramu s paletou Functions

Front Panel představuje uživatelské rozhraní celého programu. Zde se zobrazují vstupní i výstupní proměnné různých datových typů (*Integer*, *String*, *Boolean*), dále grafy a jiné prvky, jako sloupcové a ručkové indikátory. Všechny prvky, které lze umístit na *Front Panel* jsou dostupné na paletě *Controls* (pravé tlačítko myši nebo *View/Controls Palette*).



Obr. 2: Okno Front Panelu s paletou Controls

aktuální polohy kurzoru. [1]



Obr. 3: Tools paleta

2. Rozhraní pro přenos dat pro LabVIEW

Potřeba přenosu dat se objevuje už od prvních počítačů. Umožňuje komunikaci dvou vzájemně propojených PC, připojení více PC do společné sítě a sdílení dat mezi počítači. S rozvojem přístrojů pro měřicí techniku se začala komunikační rozhraní uplatňovat i v tomto odvětví. Nejprve u samostatných měřicích senzorů, kde díky převodu naměřené hodnoty na standardizovaný signál bylo možné zvětšit vzdálenost mezi senzorem a vyhodnocovací jednotkou, později i u složitějších měřicích zařízení. Zde se komunikační rozhraní uplatňuje převážně pro automatizovaný sběr dat a nastavování zařízení.

Prostředí LabVIEW má zabudovanou podporu pro nejrozšířenější komunikační rozhraní, jako je USB, sériová linka, GPIB, atd. a pomocí rozšiřujících karet je možné pracovat s průmyslovými sběrnici, například se sběrnici CAN.

2.1. Sériové komunikační rozhraní COM

Sériové komunikační rozhraní (COM port) se používá jako komunikační rozhraní u počítačové techniky a další elektroniky. Z názvu vyplývá jeho hlavní použití – připojení zařízení, která umožňují počítači komunikovat s jinými počítači, sítěmi nebo periferiemi (modemy, tiskárny, plottery).



Obr. 4: Připojovací kabel RS-232 s konektorem CANON9 [10]

Port umožňuje pouze asynchronní přenos dat, má vyčleněné linky pro sériové vysílání, příjem dat a sadu linek pro signály řízení a sledování stavu. Původní varianta portu umožňovala pouze softwarově řízené přenosy dat, to znamenalo, že procesor se přímo podílel na přenosu dat a k přenosu každého bajtu musel provést několik instrukcí, což velmi podstatně vytěžovalo procesor, zvláště pak při vysokých přenosových rychlostech. Moderní porty mají několik vyrovnávacích FIFO pamětí a využívají pro přenos dat kanálu DMA (Direct Memory Access – přímý přístup do paměti. V praxi to znamená, že jsou data z operační paměti přímo přenášena na vstupně/výstupní porty bez toho, aby procházela procesorem a tím mu umožňují vykonávat při přenosu dat i jiné instrukce).

COM port pro přenos dat využívá standartu RS-232C, který definuje, jakým způsobem se přenáší sekvence bitů a nezabývá se vyššími vrstvami komunikace. Standart dále definuje velikosti signálu, pro přenos se využívá negativní logiky. Logické jedničky odpovídá napětíová hodnota od -12 V do -3 V a logické nule napětíová hodnota od 3 V do 12 V. Pásmo mezi -3 V a 3 V je zakázané pásmo. Všechny hodnoty signálu jsou brány relativně ke společnému zemnímu vodiči. COM port není galvanicky oddělen od připojeného zařízení, uzemněné obvody počítače jsou přímo spojeny s uzemněnými obvody připojeného zařízení.

Při přenosu se přenáší data bajt po bajtu a oba směry komunikace mají oddělené linky. Přenášená informace (bajt) se skládá z:

- a) **start bitu** – ten přijímači dává informaci o zahájení přenosu dat,
- b) **datových bitů** – ty představují přenášenou užitečnou informaci,
- c) **paritního bitu** – volitelný bit, slouží k detekci chyb v přenášeném slově,
- d) **stop bitu** – ukončuje bajt a zajišťuje prodlevu mezi jednotlivými přenosy.

Po takto přenesené informaci následuje čekání na start bit dalšího bajtu. Protože se jedná o asynchronní režim přenosu dat, start bit může přijít v libovolném okamžiku, a tím pádem dochází mezi přenosem jednotlivých bajtů k různě dlouhým prodlevám.

Tento způsob přenosu informace vyžaduje, aby bylo stejné nastavení linky na obou jejích koncích. Jedná se zejména o rychlost přenosu dat, délku datových bitů, informaci o přítomnosti paritního bitu a typu parity, počet stop bitů a způsob řízení toku dat. Řízení toku dat se uplatňuje v případě, kdy by mohlo dojít k tomu, že by vysílač předbíhal přijímač, tzn. pokud by byla data vysílačem odesílána rychleji, než by je

přijímač stihl zpracovat, například díky nižšímu výkonu na straně přijímače. Nastane-li situace, kdy v polovině přijímaného bajtu nemá přijímač kam uložit celou přijímanou informaci, vyšle signál odesílateli, který okamžitě přeruší odesílání a čeká na signál, že je přijímač schopen přijmout další bajt. Po obdržení této informace přijímač ignoruje poškozený bajt a odesílatel odešle tento poslední bajt znovu. Řízení toku dat je možné realizovat hardwarově nebo softwarově. Hardwarová realizace využívá linky CTS (Clear to Send). K odeslání dalšího bajtu dojde až po nastavení CTS. Pokud má dojít k zastavení odesílání dat u softwarového řízení toku dat, přijímací strana vyšle speciální bajt (XOFF). Po obdržení bajtu se pozastaví odesílání, když přejde přijímač do stavu připravenosti, odešle bajt XON, po jehož přijetí se obnoví odesílání. Tato metoda je výrazně pomalejší než hardwarové řízení, protože se musí uvažovat čas potřebný k odeslání, přenosu, příjmu a identifikaci speciálního bajtu. Mezitím, ale stále dochází k odesílání dat, proto musí být přijímač vybaven dostatečně velkou vyrovnávací pamětí a musí umět včas zareagovat na docházející místo v této paměti. Hardwarové řešení sice vyžaduje linku pro signál CTS, ale jeho reakční doba je téměř okamžitá.

Výhodou sériové linky RS-232 je její jednoduchost, díky čemuž je vhodná pro velké množství i amatérských aplikací, využívající připojení k PC. Nevýhodou jsou nízké přenosové rychlosti, maximální délka vodičů, kterou stanovuje norma na 15 metrů a nízká odolnost proti rušení.

Ovládání sériové linky RS-232 přes LabVIEW je snadné a vychází z jednoduchosti tohoto rozhraní. Prvním krokem je konfigurace sběrnice. Zde je třeba nastavit parametry linky podle specifikací připojovaného zařízení – rychlost přenosu dat, délku sekvence bitů a paritní bit, počet stop bitů, případně způsob řízení toku dat. Při špatně nastavených parametrech linky je pravděpodobné, že LabVIEW nebude schopno se zařízením komunikovat. Po konfiguraci linky je možné data ze sériové linky RS-232 číst nebo naopak zapisovat. Na konci přenosu dat je potřeba uzavřít komunikaci na dané lince. [2] [3]

2.2. USB

USB (z anglického Universal Serial Bus; česky: univerzální sériová sběrnice) je další z rodiny sériových rozhraní. Jedná se o nástupce RS-232, a v dnešní době nejrozšířenější rozhraní pro připojení periférií k PC.



Obr. 5: Připojovací USB 2.0 kabel [11]

Přenos dat po USB sběrnici probíhá opět sériově, s tím rozdílem, že obousměrná komunikace je realizována pouze jednou dvojicí signálních vodičů, označovaných DATA+ a DATA- (D+ a D-). Jedná se o rozdílové signály, s napětovými úrovněmi 0 V a 3,3 V. Celá sběrnice je realizována čtyřmi vodiči, zbylou dvojici tvoří napájecí vodiče (+5 V U_{CC} , GND). Díky těmto vodičům je možné jednoduše zařízení napájet přímo ze sběrnice. Zařízení může při startu odebírat proud maximálně 100 mA, pokud mu to nestačí, může požádat nadřazený systém o zvýšení dodávaného proudu. V případě, že jsou pro to dostupné prostředky, systém může zvýšit proud do zařízení až na 500 mA, což je maximální hodnota proudu, kterou sběrnice umožňuje vést. Zvyšování proudu je realizováno spínáním výkonových MOSFET tranzistorů po 100 mA.

Kabel USB je složen z dvojice kroucených signálních vodičů, dvojice přímých napájecích vodičů a hliníkové fólie, sloužící jako stínění celého kabelu. Oproti rozhraní RS-232 dosahuje USB mnohonásobně větších přenosových rychlostí a rozlišuje dvě rychlosti přenosu – rychlá (full-speed) = 12Mb/s, a pomalá (low-speed) = 1,5Mb/s. Zařízení s pomalým přenosem mohou být například klávesnice nebo myš, u těchto zařízení neplatí požadavek na kroucenou signální dvojlinku, ta může být přímá. Hlavním poznávacím znamením zařízení s pomalým přenosem je kabel napevno připojený k zařízení. Veškerá zařízení, která se připojují připojovacím kabelem, tzn. takovým, který má na obou stranách konektor, jsou zařízení s rychlým přenosem dat.

Přenášená informace rozdělí do paketů, ty mohou mít dvě podoby – menší obsahují osm bajtů, větší jsou tvořeny až 256 bajty. Pakety se sdružují do rámců (frame) a přenos dat se poté realizuje pomocí 1 ms trvajícího rámu. Jeden rám může obsahovat malé i velké pakety pro několik připojených zařízení. U více připojených zařízení je za

rozdělení paketů pro jednotlivá zařízení zodpovědný rozbočovač sběrnice (hub). Malý paket je určen pro pomalé zařízení, velký paket pro rychlé zařízení. Ke kódování informace se používá metoda NRZI (Non Return To Zero), při které nuly vedou ke změně úrovně a jedničky žádnou změnu úrovně nevyvolají. Každý paket obsahuje jeden počáteční (zaváděcí) bajt, sloužící zařízením k synchronizaci přenosu, protože sběrnice neobsahuje žádný synchronizační hodinový signál. Tento bajt má hodnotu 00000001b, kterou po zakódování zařízení identifikuje jako osm změn logické úrovně.

Obvody pro příjem i vysílání dat jsou realizovány v jedné součástce. USB dále obsahuje jednotku SIE (Serial Interface Engine), která je doplněna vyrovnávací FIFO pamětí. Jednotka se stará o veškerou komunikaci a podstatně ulehčuje práci se sběrnici. Do FIFO paměti stačí zapsat informace, které se mají odeslat a SIE zařídí veškeré ostatní operace. Stejně je to i se čtením, stačí vyčíst obsah FIFO paměti. Jednotka SIE dnes zpravidla bývá součástí řadiče USB.

Sběrnice umožňuje připojení až 127 zařízení najednou, číslo je dáno sedmibitovým registrem zajišťujícím adresaci zařízení. Adresu 0 má vždy centrální prvek, nejčastěji se jedná o PC. Sběrnice je zapojena do stromové struktury, kdy v kořeni je centrální řídicí zařízení a za ním se sběrnice dále větví. K rozdělení portů se používá rozbočovač (hub). I ten je adresován, proto je maximální počet připojených zařízení menší o počet použitých rozbočovačů. Při více připojených zařízeních se celková přenosová rychlost dělí mezi jednotlivá zařízení, ale i při připojení jejich maximálního počtu je rychlost, při porovnání s RS-232, mnohem vyšší.

Sběrnice USB je typu Plug & Play, to umožňuje připojovat nebo odpojovat zařízení „za provozu“ bez nutnosti restartování počítače nebo ručního zavádění ovladačů. Po připojení je zařízení během několika okamžiků připraveno k práci, pokud bylo zařízení připojeno poprvé, operační systém většinou sám vyhledá ovladače zařízení, případně může uživatele vyzvat k instalaci vhodných ovladačů, pokud vhodné ovladače nenalezne.

Výhodou USB sběrnice je její rychlost, možnost napájet připojená zařízení a její rozšíření do veškeré počítačové techniky a elektroniky. Nevýhodou je vyšší technická náročnost. Při použití USB sběrnice je potřeba na straně zařízení USB řadič, který zajišťuje komunikaci s PC a na straně PC ovladač. Problémem je, že jak ovládání řadiče, tak ovladač v PC jsou složité programy, nemluvě o problému přenositelnosti

ovladače mezi různými operačními systémy, které amatérskému uživateli znepříjemňují použití USB sběrnice pro měření nebo řízení jeho aplikací.

Velké množství moderních měřicích zařízení, např. multimetry, generátory signálu, osciloskopy, atd., má vyvedeno USB a je možné je připojit k PC. Většina výrobců k těmto zařízením dodává vlastní software a pro mnoho z nich jsou k dispozici ovladače pro LabVIEW. Není proto žádný problém používat pro sběr a vyhodnocení naměřených dat právě LabVIEW. Po připojení přístroje a otevření komunikačního kanálu mezi ním a počítačem můžeme okamžitě provádět zápis nebo čtení hodnot, jejich zpracování, nastavování parametrů a jiné. [2] [4]

2.3. Rozhraní GPIB

GPIB (z anglického General Purpose Interface Bus; česky: rozhraní univerzální sběrnice), původním názvem HP-IB (Hewlett Packard Interface Bus), je digitální sběrnice poprvé použitá v roce 1972 společností Hewlett-Packard pro jednodušší komunikaci mezi zkušebními a měřicími přístroji. V roce 1975 bylo rozhraní standardizováno pod označením IEEE 488. Dnes je to nejrozšířenější komunikační rozhraní pro měřicí techniku na světě.



Obr. 6: Kabel GPIB [12]

Jedná se o 24 vodičovou paralelní sběrnici. Data jsou přenášena po bajtech pomocí osmi datových vodičů (DI01 – DI08). Tři další linky slouží pro řízení přenosu dat (DAV, NRFD, NDAC), pět vodičů je vyhrazeno pro řízení rozhraní (ATN, IFC, REN, SRQ, EOI) a posledních osm vodičů je zemnicích (GND), ty uzavírají proudové smyčky a slouží jako stínění. Přenos dat probíhá asynchronně s hardwarovým řízením. Při přenosu se využívá negativní TTL logika – logické nule odpovídá napěťová hodnota

$> 2,1 \text{ V}$; logické jedničky hodnota $< 0,8 \text{ V}$. Maximální přenosová rychlost sběrnice je 1 MB/s , v praxi jsou ale maximální hodnoty přenosu okolo $250 - 500 \text{ kB/s}$. Původní varianta sběrnice počítá s rozdělením zařízení do třech skupin:

- a) **řídící člen** – zodpovídá za řízení přenosu dat a řízení rozhraní (počítač, GPIB karta),
- b) **mluvčí** – zařízení, které posílá data po sběrnici (generátor, osciloskop),
- c) **posluchač** – zařízení, které přijímá data (tiskárna).

Sběrnice používá pro připojená zařízení 5 bitové adresování, z tohoto důvodu může být k jedné kartě připojeno maximálně 30 logických zařízení. Fyzické zařízení má zpravidla funkce mluvčího i posluchače, dochází tak ke snížení počtu zařízení na polovinu. Řídící člen se stará o to, aby v jednom okamžiku mělo funkci mluvčího přidělenou pouze jedno zařízení, ostatní přístroje jsou v režimu poslechu. Je možné mít k jedné sběrnici připojeno více zařízení s funkcí řídicího členu, ale stejně jako v případě mluvčího, v jednom okamžiku může pouze jedno ze zařízení řídit přenos dat a ovládat rozhraní.

V původní verzi protokolu bylo při přenosu dat nutné využívat tři linky pro řízení přenosu dat, tzv. handshaking. Právě tato nutnost způsobovala nízké přenosové rychlosti. V roce 2003 přišla firma National Instruments s novým, zpětně kompatibilním, protokolem označovaným HS-488 (IEEE-488.1-2003), který upravuje pravidla při přenosu dat a odstraňuje požadavek na používání signálů DAV, NRFD, NDAC. Tím dochází ke zvýšení maximální rychlosti přenosu dat na hodnotu 8 MB/s . Při zapojení přístrojů s různými rychlostmi na jednu sběrnici je rychlost sběrnice dána rychlostí nejpomalejšího přístroje. Zvláštností sběrnice je používaný typ konektoru. Využívá se dvou-hlavový konektor se samčím konektorem na jedné straně a samičím na straně druhé, to umožňuje jejich snadné stohování.

Výhodou sběrnice GPIB je jednoduché připojení více zařízení na jedinou sběrnici, její zavedenost a velká podpora ze strany výrobců. Nevýhodou může být mohutná konstrukce konektoru a kabelu a v porovnání s USB (nebo FireWire) sběrnici omezená rychlost a vysoká cena.

V LabVIEW probíhá programování GPIB sběrnice podobně jako u sériového rozhraní. Po otevření portu (adresy připojeného zařízení) můžeme okamžitě provádět zápis nebo čtení hodnot a jejich další zpracování. [5] [6] [7]

2.4. Speciální měřicí karty

Další možností, jak připojit k LabVIEW různé měřicí nebo řízené periferie, jsou speciální měřicí karty. Tyto rozšiřující karty obsahují digitální a analogové vstupy a výstupy, se kterými je možné pomocí speciálních funkcí v LabVIEW pracovat. Existuje mnoho variant karet, mezi jejich hlavní rozdíly, mimo ceny pohybující se od jednotek tisíc až po statisíce korun, patří rozlišení a rychlost A/D a D/A převodníků, počet jednotlivých vstupů, výstupů a způsob připojení k nadřazenému systému. Firma National Instruments dodává na trh širokou škálu měřicích karet, jak do PCI slotu, tak do USB portu počítače.

Zástupcem rozšiřující měřicí karty do PCI slotu počítače může být karta PCI-6013, jedna z nejzákladnějších karet z nabídky NI. Karta má 16 analogových vstupů se 16 bitovým A/D převodníkem a rychlostí 200 kS/s, 2 analogové výstupy se 16 bitovým D/A převodníkem a rychlostí 10 kS/s. Rozsah analogových vstupů a výstupů je ± 10 V. Dále má karta osm digitálních vstupů/výstupů a dva čítače/časovače. Výstupem z PCI karty je 68 pinový konektor VHDCI (z anglického Very-High-Density Cable Interconnect; česky propojovací kabel s vysokou propustností dat), proto je ještě vhodné doplnit kartu svorkovnicí.



Obr. 7: Měřicí karta NI PCI-6013 [13]

Seznam všech aktuálně připojených zařízení i měřicích desek je možné získat pomocí nástroje *Measurement & Automation Explorer* od firmy National Instruments, který se instaluje současně s vývojovým prostředím LabVIEW. Seznam nalezneme v levém sloupečku *My System* pod záložkou *Devices and Interfaces*. Jsou zde vypsána zařízení, jejich adresy a v případě sběrnice RS-232 a GPIB je zde funkce pro vyslání příkazu na sběrnici, který může sloužit pro otestování připojeného zařízení. [8]

3. Možnosti přístupu k LabVIEW programů přes webové rozhraní

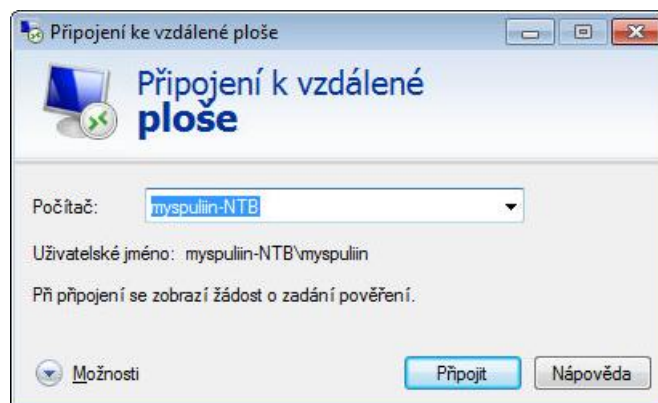
Schopnost řídit aplikaci bez nutnosti být fyzicky u počítače, který má řízení na starosti, je v dnešní době stále více žádaná funkce systémů. Mnoho průmyslových procesů je řízeno, popřípadě sledováno na dálku z řídicího střediska. Na tomto principu je řízena doprava, distribuce elektrické energie ke koncovým uživatelům i samotná výroba elektrické energie, například ve vodních elektrárnách, kdy je na elektrárně přítomen dozor, ale veškeré ovládání je realizováno ze vzdáleného střediska centrálního řízení.

Tento způsob vzdálené správy se rozrůstá i do dalších, méně důležitých odvětví, například měření na unikátním laboratorním vybavení, které je dostupné pouze na specializovaném pracovišti. V těchto případech se vzdáleného řízení využívá ne z důvodu eliminace chyby jednotlivce nebo potřeby centrálního řízení, ale protože tato měření pak mohou provádět i osoby, které nemají fyzický přístup na pracoviště.

3.1. Připojení ke vzdálené ploše

Možností, jak docílit vzdálené správy PC, je několik. Samotný operační systém Microsoft Windows 7 má v sobě implementovanou funkci *Připojení ke vzdálené ploše*. Bohužel, tato funkce není použitelná na všech edicích systému Windows 7, respektive ze všech edic systému je možné se připojit ke vzdálené ploše, ale pouze pokud je na počítači, ke kterému se připojujeme, nainstalována edice systému Windows 7 Professional a vyšší (Windows 7 Ultimate, Windows 7 Enterprise). V operačním systému Windows se funkce vzdálené plochy objevuje od verze systému Windows XP, dostupná je opět pouze ve vyšší verzi Windows XP Professional.

Připojení ke vzdálené ploše je velmi jednoduché. Po spuštění aplikace *Připojení ke vzdálené ploše* se zobrazí dialogové okno, vyplníme název počítače, ke kterému se chceme připojit, případně jeho IP adresu, dále uživatelské jméno účtu, k němuž přistupujeme, a heslo. Při správné kombinaci dojde ke spojení, na počítači, ke kterému se připojujeme, se zobrazí přihlašovací obrazovka. Pokud se na tomto počítači uživatel přihlásí, dojde k přerušení spojení se vzdálenou plochou, jinak je možné počítač plně ovládat přes internetový protokol pomocí připojeného počítače. V tomto režimu je možné spouštět aplikace, zobrazovat, měnit a mazat soubory a ovládat počítač tak, jako bychom u něho fyzicky byli.



Obr. 8: Připojení ke vzdálené ploše

Jednoznačnou výhodou tohoto přístupu je skutečnost, že operační systém má pro tuto potřebu připravenou funkci. Pokud se ovšem bude chtít ke vzdálené počítači připojit uživatel, využívající jiný operační systém než od firmy Microsoft, například Mac OS X, případně Linux, nebude toto spojení možné. Další obtíže nastanou, pokud bude ovládaný počítač „skryt“ v síťové struktuře nebo bude připojen do zvláštní (podnikové) domény. Nejlepším řešením těchto negativ spojení je mít vyhrazen počítač pouze pro účely vzdáleného připojení, pokud možno s veřejnou, statickou IP adresou a bez zařazení do firemní sítě, domény. Poslední důležitou věcí, kterou je třeba si uvědomit, je to, že kdokoli, komu poskytneme údaje pro vzdálené připojení, bude mít veškerou kontrolou nad počítačem, bude moci libovolně prohlížet a měnit data a nastavovat funkce počítače. To může být potenciálně nebezpečné.

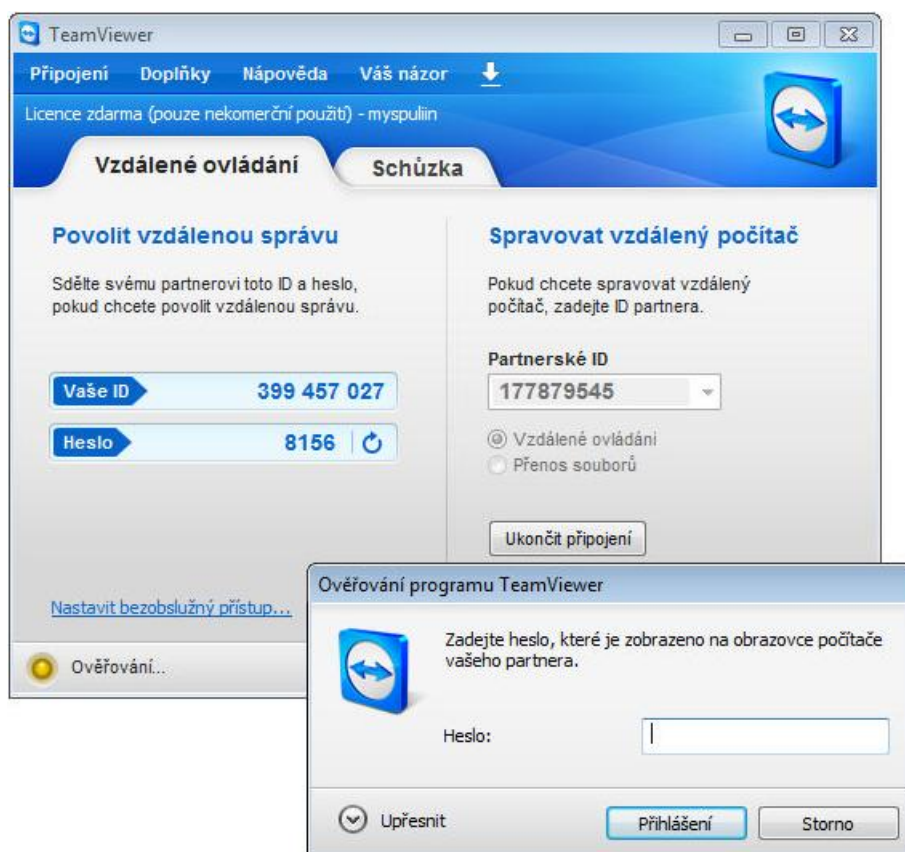
3.2. Aplikace TeamViewer

Další možností je využití aplikace třetích stran. Velmi oblíbenou aplikací a to i v komerční sféře je aplikace *TeamViewer*. Tato aplikace umožňuje kompletní vzdálený přístup k hostitelskému počítači. Díky její jednoduchosti a funkčnosti je aplikace velmi oblíbená mezi počítačovými administrátory a správci, protože umožňuje řešení základních softwarových problémů na dálku.

Stejně jako *Připojení ke vzdálené ploše* je možné tuto aplikaci použít ke spouštění aplikací na vzdáleném počítači. Tím, že se jedná o aplikaci třetí strany, odpadá problém se vzájemnou kompatibilitou operačního systému řízeného a řídicího počítače. Aplikace *TeamViewer* je vyvíjena pro všechny moderní operační systémy – Microsoft Windows, Mac OS X, Linux a dokonce i pro mobilní platformy Mac iOS a Google Android.

Po jednoduché instalaci, kde je třeba určit instalační adresář a budoucí využití programu – pro nekomerční účely je program dostupný zdarma, se v hlavním okně

programu zobrazí dvě podokna. V prvním je uvedeno ID a heslo pro případ, že se bude nějaký uživatel připojovat k našemu počítači. Druhé podokno je připraveno pro naše připojení k jinému počítači, v tomto případě potřebujeme znát ID a heslo druhého počítače. V případě úspěšného spojení se zobrazí plocha partnerova počítače v okně aplikace *TeamViewer*. Opět máme plnou kontrolu nad řízeným počítačem, ovšem při použití této aplikace se vyhneme, na rozdíl od funkce *Připojení ke vzdálené ploše*, problémům s kompatibilitou operačních systémů, dále se zde nevyskytuje problém s IP adresou, která je dnes ve velkém množství případů dynamická, protože *TeamViewer* používá pro identifikaci kombinaci jedinečného ID a hesla.



Obr. 9: Připojení ke vzdálenému PC pomocí aplikace TeamViewer

Mezi další funkce, které *TeamViewer* podporuje, patří možnost vizuálního záznamu po dobu připojení, hlasový a textový chat mezi uživateli na obou stranách, integrovaný správce souborů, který umožňuje jednoduší správu, procházení a kopírování souborů a složek na lokálním a vzdáleném počítači. Stejně jako v případě *Připojení ke vzdálené ploše* je třeba brát v potaz to, že připojený uživatel má plný přístup k počítači.

3.3. LabVIEW Remote Panel

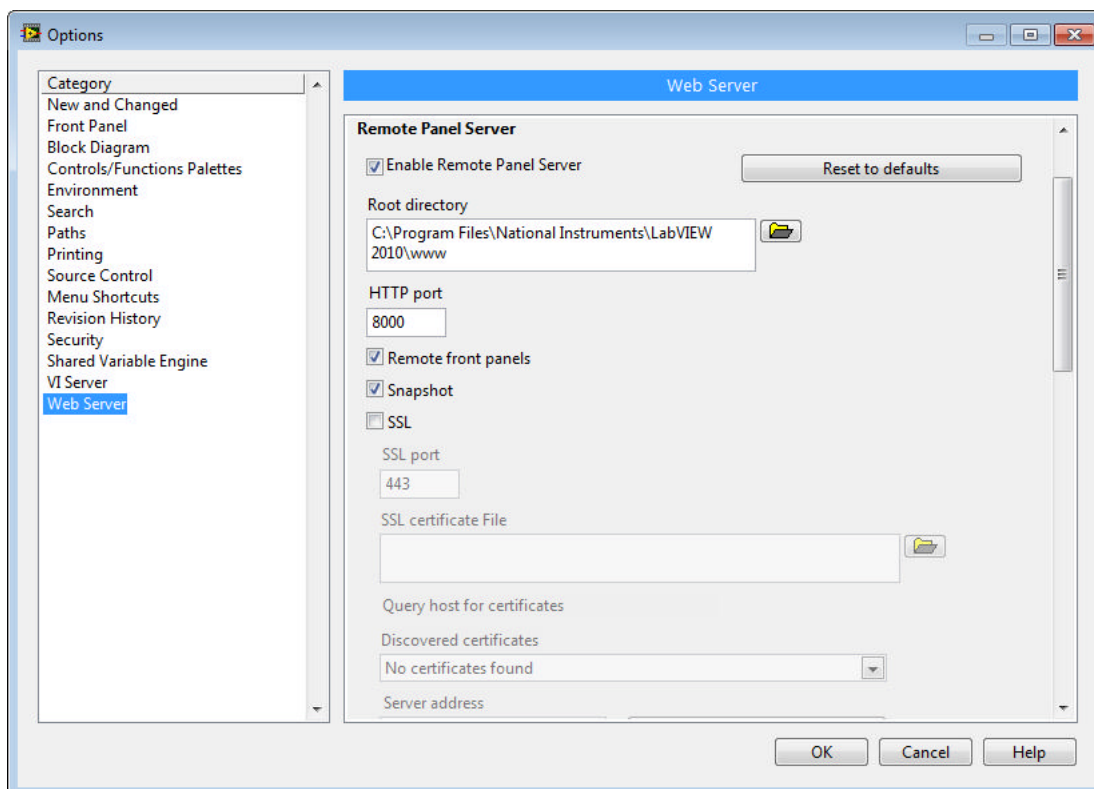
Dvě poměrně zajímavé funkce pro vzdálený přenos, případně řízení a ovládání nabízí přímo LabVIEW. Funkce *DataSocket Technology* umožňuje okamžité sdílení naměřených hodnot mezi více LabVIEW programy na lokálním počítači nebo mezi počítači v síti. Toho můžeme využít například při získávání hodnot z měřicího čidla. Naměřené hodnoty mohou být zpracovány na lokálním počítači, být dále přeneseny po síti k dalšímu zpracování a v tom stejném okamžiku dostupné na webové stránce. Příjem i odesílání dat je v programu realizováno samostatnými funkčními bloky, jejichž vstupem je URL adresa (z anglického Uniform Resource Locator; česky: jednotný lokátor zdrojů), definující konkrétní komunikační protokol. DataSocket Technology podporuje protokoly:

- a) **DSTP (DataSocket Transport Protokol)** – jedná se o nativní protokol LabVIEW určený přímo pro *DataSocket Technology*. Při použití tohoto protokolu probíhá komunikace přes DataSocket Server.
- b) **OPC (OLE for Process Control)** – protokol speciálně navržený pro sdílení dat v reálném čase, v průmyslu využíváný ke sdílení dat generovaných automatizovanými systémy. K běhu je opět nutný server, v tomto případě *OPC server*.
- c) **FTP (File Transfer Protokol)** – protokol určený ke čtení ze vzdáleného úložiště, *DataSocket Technology* neumožňuje zápis na FTP server.

Pro kompletní ovládání běžícího programu se více než *DataSocket Technology* hodí funkce *Remote Panel*. Ta umožňuje zobrazovat a ovládat *Front Panel* na dálku pomocí internetového protokolu. LabVIEW má pro tento případ zabudovaný web server, který zajišťuje komunikaci mezi připojenými počítači. Funkce vyžaduje k běhu LabVIEW nebo LabVIEW Run-Time Engine, který je volně ke stažení ze stránek NI a u kterého vzdálená správa probíhá pomocí okna webového prohlížeče. Potřebné zásuvné moduly pro nejpoužívanější webové prohlížeče Internet Explorer a Mozilla Firefox (Google Chrome zatím není ze strany National Instruments a LabVIEW podporován) se do počítače instalují spolu s LabVIEW Run-Time Enginem.

LabVIEW Web Server se nastavuje v okně *Tools/Options/Web Server*. Ke spuštění dojde po zatržení *Enable Remote Panel Server*. V nabídce je možné nastavit konkrétní HTTP port, povolit snímání obrazovky a zapnout šifrování pomocí SSL, opět s definicí příslušného SSL portu, serveru a certifikátu. LabVIEW Web Server umožňuje, aby byly

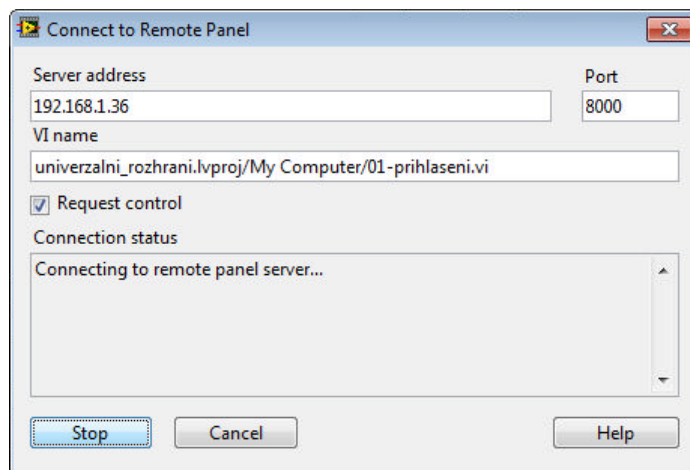
připojeným uživatelům viditelné pouze některé soubory VI, ty jsou vybírány v části *Visible VIs*, kde je možné definovat i časový limit připojení. Ten najde uplatnění v případě, že je k serveru připojeno více uživatelů (verze LabVIEW Professional nabízí možnost až 5 připojených uživatelů k serveru naráz). Po uplynutí časového limitu se řízení aplikace předá dalšímu uživateli. V poslední části okna se nastavuje povolení přístupu. Přístup se dá povolit všem uživatelům, pouze konkrétním IP adresám nebo naopak určitým IP adresám, případně rozsahům adres přístup odepřít.



Obr. 10: Okno nastavení LabVIEW Web Serveru

Připojení počítače k LabVIEW Web Serveru s nainstalovaným LabVIEW se provádí v úvodním okně *Getting Started*. Po kliknutí na *Operate/Connect to Remote Panel* se otevře nové okno, zde je třeba vyplnit adresu serveru, která je akceptována v podobě IP adresy nebo názvu počítače. Ve vedlejším políčku se specifikuje HTTP port, stejný jako je nastavený na straně serveru. Určitou zvláštností je to, že ve výchozím nastavení je číslo portu na serveru 8000, ale na straně uživatele je předvyplněné číslo portu 80. Je potřeba, aby byla čísla portů stejná, jinak bude pokus o spojení neúspěšný. Do posledního políčka se wpisuje jméno VI souboru, ke kterému se chceme připojit a který je na serveru spuštěn. V případě, že je spuštěno samotné VI, bez zařazení do projektu, stačí do políčka vepsat jméno VI, například: *MyVI.vi*. Pokud je VI součástí projektu, musí se v názvu specifikovat, o jaký projekt se jedná, syntaxe zápisu

takového jména souboru může vypadat: *MyProject.lvproj/My Computer/MyVI.vi*. Nakonec je ještě možné zatrhnout volbu *Request Control*, která zajistí, že ihned po připojení bude počítači předáno řízení programu. To ale platí pouze v případě, že program neřídí jiný uživatel.



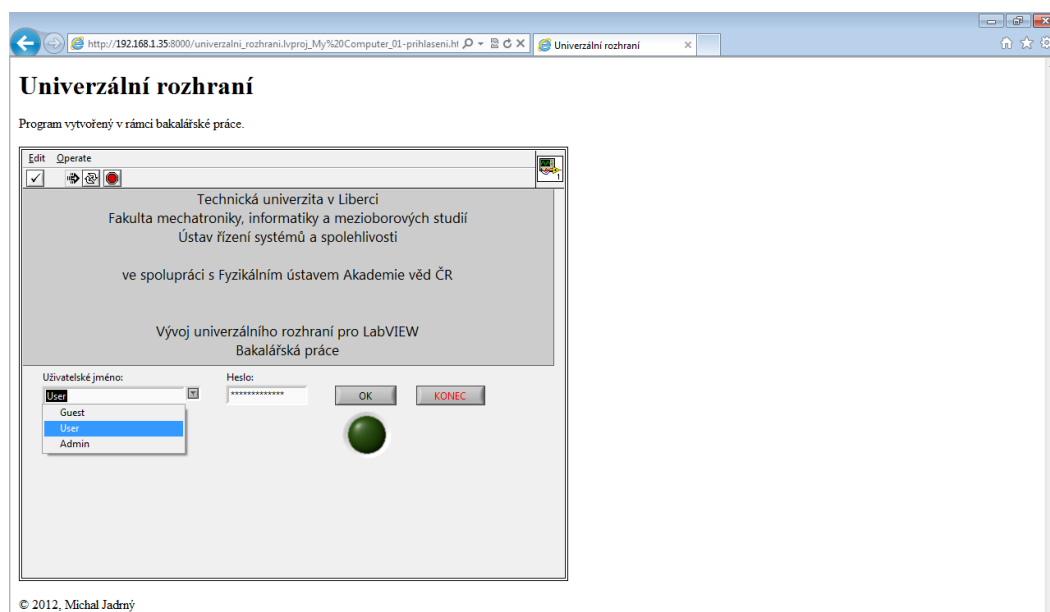
Obr. 11: Připojení ke vzdálenému VI souboru pomocí LabVIEW

U připojení počítače k LabVIEW Web Serveru přes webový prohlížeč je nastavení složitější. Mimo běžného nastavení serveru tak, jak je popsáno výše, se ještě musí vytvořit speciální HTML soubor, který se zobrazí vzdáleně se připojujícímu uživateli a který zajistí, že se na počítači spustí Run-Time Engine. K tomuto účelu slouží HTML tagy *<object>* a *<embed>*. LabVIEW má připraveného průvodce, který daný soubor generuje. Průvodce je dostupný v nabídce *Tools/Web Publishing Tools*. V prvním okně se vybere dané VI, ke kterému se bude uživatel připojovat pomocí webového prohlížeče, a zobrazovací režim. Na výběr jsou tři druhy:

- a) **Embedded** – chová se jako normálně spuštěný Front Panel, je možné ho pomocí prohlížeče ovládat,
- b) **Snapshot** – statický obrázek zobrazený na webové stránce, není možné ho vzdáleně ovládat,
- c) **Monitor** – statický obrázek s pravidelným obnovováním, není možné ho vzdáleně ovládat.

V dalším okně se zadává titulek stránky, text zobrazený nad *Front Panelem* a text zobrazený pod *Front Panelem*. V posledním okně je možné vybrat lokální adresář, ve kterém je HTML soubor uložen, případně ještě upravit jeho název. Kliknutím na tlačítko *Save to Disk* se vygeneruje soubor a zobrazí odkaz, například: http://192.168.1.34:8000/univerzalni_rozhrani.lvproj_My%20Computer_01-prihlaseni.html.

Pomocí tohoto odkazu se realizuje připojení uživatele k web serveru. V obou případech připojení k LabVIEW Web Serveru musí být na straně serveru otevřen a spuštěn daný VI soubor.



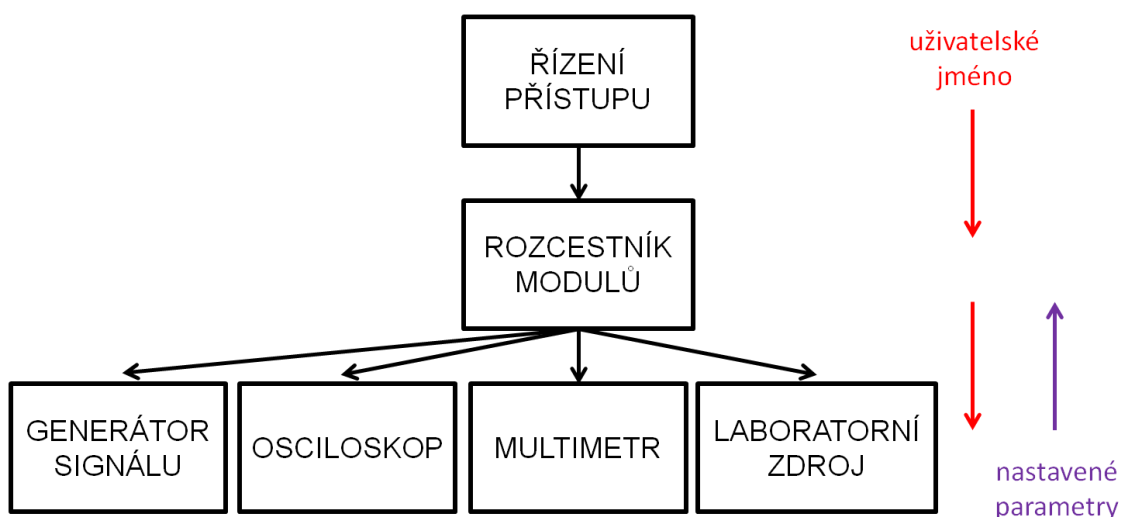
Obr. 12: Připojení ke vzdálenému souboru pomocí webového prohlížeče

Pokud se k serveru úspěšně připojí vzdálený klient, server mu odešle pouze *Front Panel*, blokový diagram a všechny podprogramy zůstanou uloženy pouze na straně serveru a načítají se pouze v případě potřeby. Vzdálený klient má možnost pracovat s *Front Panelem* stejně, jako by byl spuštěn přímo z jeho počítače. Jednotliví uživatelé žádají o kontrolu nad programem pomocí kliknutí pravým tlačítkem myši a příkazu *Request Control*. Serverový počítač může uzavřít řízení, aby ho nikdo nemohl převzít nebo si řízení vyžádat a to je mu okamžitě předáno. Všechny tyto možnosti jsou dostupné v běžícím programu po kliknutí pravým tlačítkem myši. Všichni připojení uživatelé mají z pohledu serveru stejná práva a platí pravidlo, že kdo si dříve vyžádá kontrolu, ten ji dříve získá. Změnit pořadí uživatelů je možné na počítači, na kterém běží server. V okně *Tools/Remote Panel Connection Manager* jsou zobrazeny informace o aktuálně připojených uživatelích, datových tocích, časech řízení, atd. V nabídce se dají odpojit konkrétní uživatelé i změnit jejich pořadí.

Velkou výhodou tohoto způsobu vzdáleného přístupu je skutečnost, že uživatelům nedáváme přístup k celému počítači, ale pouze ke konkrétním souborům. Na druhou stranu, vzdálený přístup pomocí aplikace třetí strany nebo *Připojení ke vzdálené ploše* nám umožňují ladění a úpravu zdrojového souboru LabVIEW, což může určitě najít své uplatnění. [9]

4. Návrh univerzálního rozhraní

Dvě hlavní požadované funkcionality univerzálního měřicího rozhraní jsou možnost vzdáleného ovládání a obsluha modulů na základě přístupových práv přihlášeného uživatele. Toto univerzální rozhraní bude sloužit jako určitá grafická nadstavba a spojovací článek mezi několika typy modulů. Rozhraní bude zobrazovat dostupné moduly podle aktuálně připojených přístrojů, získávat a zobrazovat základní informace o nastavení modulů, případně o aktuálních měřených hodnotách a tyto informace bude logovat do souboru. V první řadě je tedy potřeba se zaměřit na systém řízení přístupu, který bude zodpovídat za identifikaci uživatele a tuto informaci bude předávat do nižších vrstev programu.



Obr. 13: Blokový diagram programu

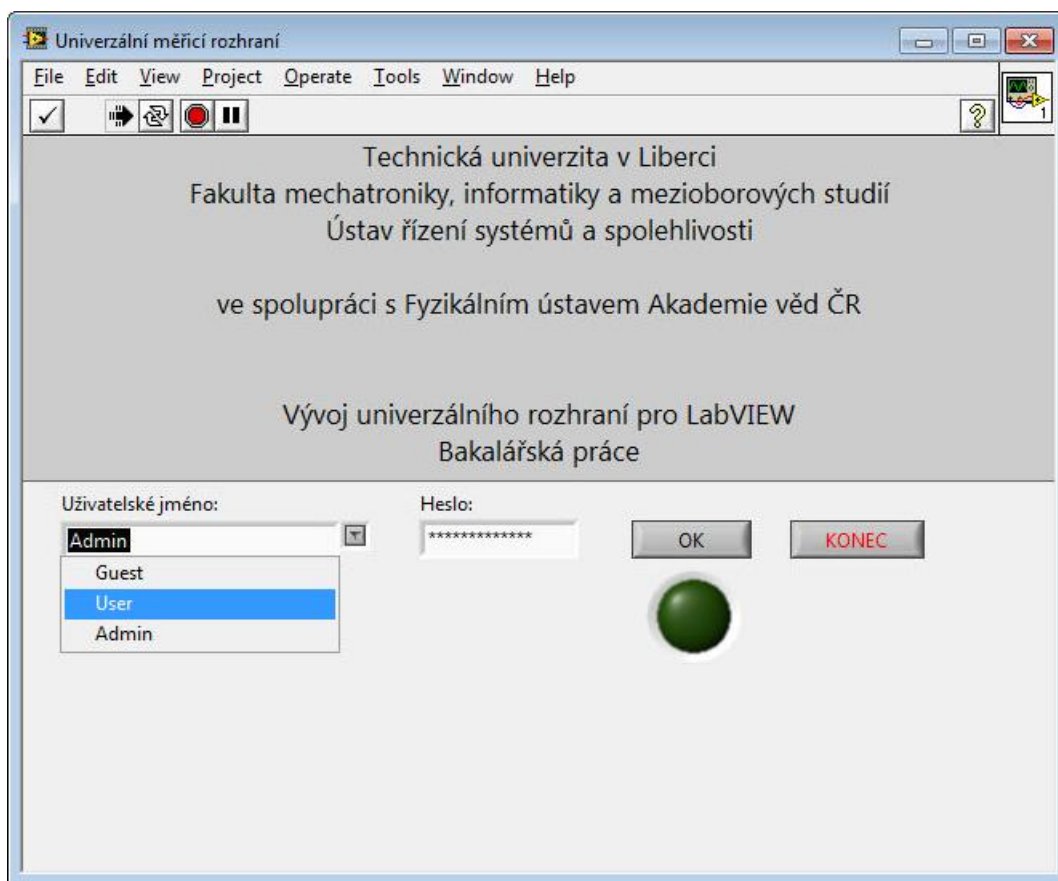
4.1. Systém řízení přístupu

Úvodní obrazovka, kterou uvidí uživatel při spuštění programu, informuje o tom, že se jedná o program vytvořený v rámci bakalářské práce na Fakultě mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci, ve spolupráci s Fyzikálním ústavem Akademie věd České republiky. Uprostřed okna je nabídka s dostupnými uživatelskými účty, ke kterým je možné se přihlásit. V programu jsou definovány tyto uživatelské účty:

- a) Guest (host)
- b) User (uživatel)
- c) Admin (administrátor)

Při výběru účtu Guest a stisknutí potvrzujícího tlačítka dojde k přihlášení uživatele s nejnižšími uživatelskými právy. Pokud je v nabídce vybrán účet User nebo Admin,

zobrazí se pole pro zápis uživatelského hesla. Při správném zadání a stisknutí tlačítka *OK* dojde opět k přihlášení uživatele. Nesprávně zadané heslo neumožní přihlášení uživatele, počet zadání hesla není nijak limitován, nehrozí žádné zablokování programu. Cílem řízení přístupu je spíše zamezit nezkušeným uživatelům nastavovat některé parametry programu a omezit jejich možnosti při ovládání modulů než ochrana programu proti neoprávněnému vstupu.



Obr. 14: Grafická podoba přihlašovacího okna

Co se týče zdrojového kódu, při spuštění programu jsou pole pro výběr uživatelského jména a hesla resetovány do defaultního nastavení, tzn. do polí jsou nastaveny prázdné řetězce. To samé platí pro sdílenou proměnnou (proměnná, která je sdílená mezi všemi soubory, které jsou součástí projektu), do které je ukládána identifikace aktuálně přihlášeného uživatele. Nakonec dojde k zavolání podprogramu *nacteni_hesla.vi*, zajišťující načtení hesla ze souboru (viz 4.2. Přístupové heslo). V dalším kroku se porovnává zadané uživatelské jméno a heslo se správnými údaji. Celý tento krok se vykonává v nekonečné smyčce *While*, k jejímu ukončení dojde při správně zadaných přihlašovacích údajích nebo při ukončení programu. Aby smyčka zbytečně nezatěžovala procesor počítače, čímž by mohla přispět k jeho nestabilitě, se

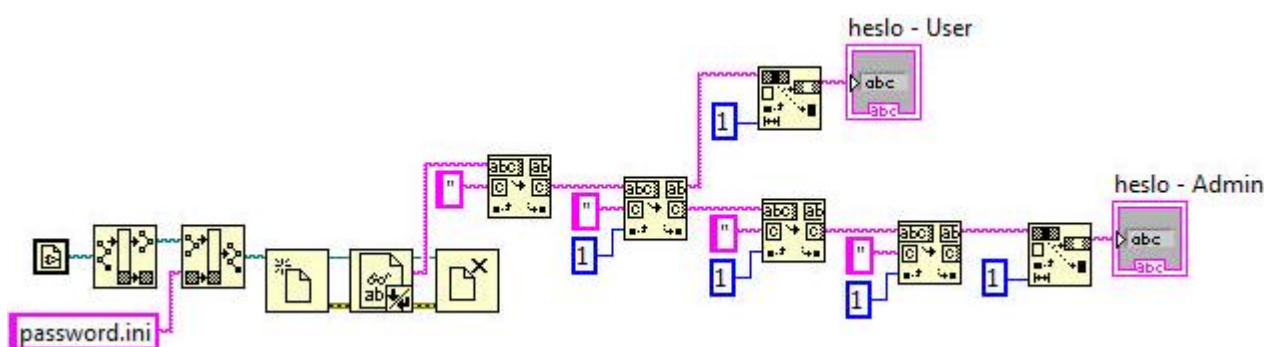
provádění smyčky uměle zpomaluje pomocí bloku *Time Delay* s hodnotou nastavenou na 200 ms. Díky tomu se smyčka provede pouze 5 krát za sekundu. Pokud dojde ke správně zadané kombinaci uživatelského jména a hesla, smyčka *While* se ukončí a přejde se na další krok. Zde se do sdílené proměnné zapíše jméno přihlášeného uživatele a otevře se nový soubor VI sloužící jako rozcestník mezi připojenými moduly. Otevření je realizováno pomocí bloku *Invoke Node*, který slouží pro práci s *Front Panelem* a umožňuje programově měnit parametry panelu, otevírat, spouštět, zastavovat a ukončovat soubory VI a jiné funkce. Vstupem do tohoto bloku je reference na soubor VI, se kterým chceme pracovat. Referenci získáme pomocí bloku *Open VI reference*, do něhož vstupuje cesta k souboru. Tento blok umí pracovat s relativním adresováním, proto stačí umístit soubor do kořenové složky projektu a místo cesty k souboru stačí uvést jeho jméno. Po otevření souboru můžeme opět pomocí bloku *Invoke Node* daný soubor spustit a poté uzavřít referenci na otevíraný soubor. Posledním krokem je uzavření přihlašovacího okna.

4.2. Přístupové heslo

Přístupové heslo pro jednotlivé uživatelské účty je získáváno ze souboru *password.ini* umístěného v kořenovém adresáři programu. To umožňuje obsluhu jednoduchou změnu hesla bez nutnosti zásahu do zdrojových souborů programu. Obsah souboru *password.ini* může být podle potřeby upravován ve smyslu změny popisků u jednotlivých položek souboru. Důležité je, aby byl zachován původní zápis hesel v uvozovkách a jméno souboru *password.ini*.

Podprogram, v programovacím prostředí LabVIEW nazýván subVI, *nacteni_hesla.vi* nejprve upraví cestu k souboru *password.ini* tak, že z cesty k podprogramu odebere název podprogramu a místo toho adresu doplní názvem souboru *password.ini*. Tento postup je zvolen z důvodu přenositelnosti programu do jiného počítače, kde může být jiná adresářová struktura. V další části podprogramu se otevírá výše zmíněný soubor, ale funkce na otevření souboru neumí pracovat s relativním adresováním souboru a vyžaduje adresu v absolutním zápisu. Jinou možností by bylo, aby uživatel při prvním spuštění programu zadal ručně nebo pomocí dialogového okna cestu k souboru. Tento postup není uživatelsky moc přívětivý, a protože není soubor nijak šifrován, celé řízení přístupu by postrádalo hlubší smysl.

Po úspěšném otevření souboru se celý jeho obsah načte do textového řetězce a soubor se uzavře. V textovém řetězci se vyhledá první uvozovací znak a vše před ním se smaže. První uvozovka, spolu se zbytkem řetězce, postupuje programem dále a hledá se další uvozovací znak. Po jeho najití se z části řetězce před ním odstraní první znak (první uvozovka) a výsledek se запиše do proměnné *heslo – User*. Ve zbytku řetězce se opět vyhledá první a druhý uvozovací znak a po oříznutí prvního znaku se výsledek запиše do proměnné *heslo – Admin*. Výsledkem po provedení podprogramu jsou dvě textové proměnné, ve kterých jsou dvě hesla načtená ze souboru *password.ini*.



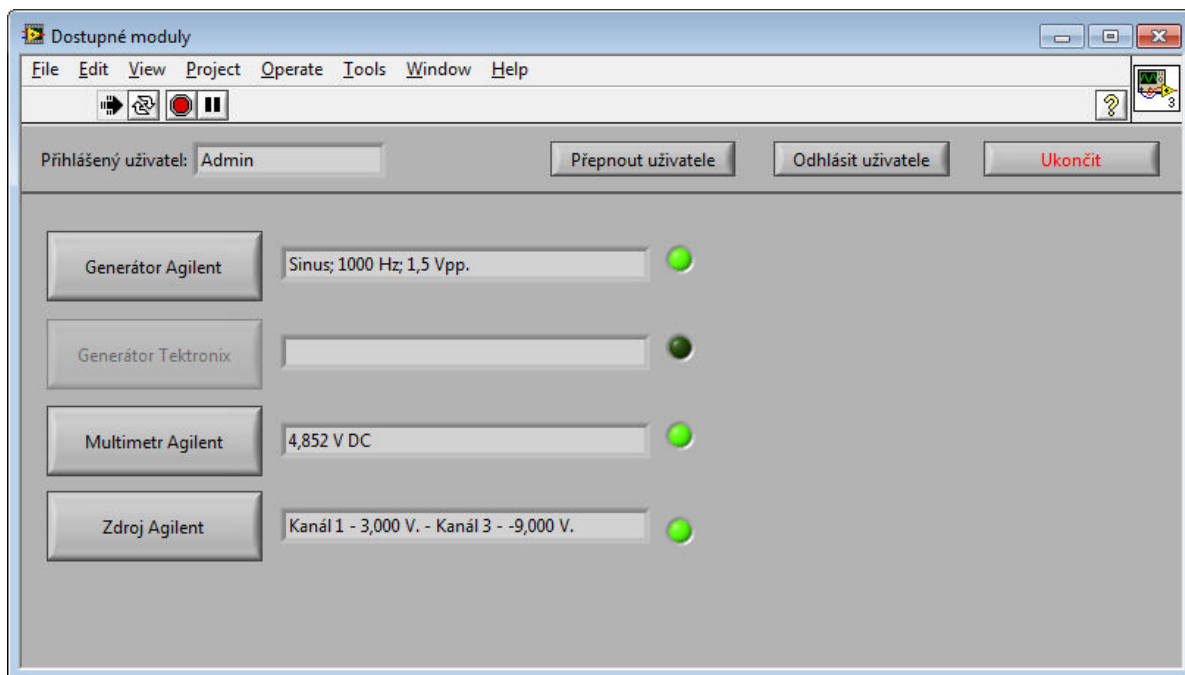
Obr. 15: Podprogram pro získání hesla ze souboru

4.3. Rozcestník modulů

Základem rozcestníku a dále potom i modulů je horní informační a ovládací lišta, která zobrazuje jméno přihlášeného uživatele, umožňuje jeho přepnutí, odhlášení, popřípadě ukončení celého programu. Při volbě ukončení programu dojde po potvrzení dialogového okna k ukončení programu. Pokud je zvolena možnost ohlášení uživatele, dojde k uzavření všech aktivních oken programu a program se vrátí na úvodní přihlašovací obrazovku. Veškeré ovládací prvky v programu i modulech jsou několikrát za sekundu aktualizovány a díky tomu je možné přepnutí uživatele za chodu programu a pravidelná aktualizace v tomto případě zajistí překreslení ovládacích prvků tak, aby odpovídaly právům nově přihlášeného uživatele. V modulech je tato funkce dobře využitelná v případě, že administrátor potřebuje nastavit například výstupní limity u připojeného zařízení a poté předat řízení procesu uživateli s nižšími přístupovými právy. Není třeba ukončovat a znovu spouštět program, stačí přepnout uživatele, provést přenastavení limitů a poté znovu přepnout uživatele.

Mimo horní lišty rozcestník obsahuje ikony jednotlivých modulů, zastupující konkrétní přístroj a dále zobrazení základních parametrů přístroje. Tato informace je

předávána z modulů na rozcestník pomocí sdílené proměnné. Rozcestník se dále stará o logování těchto parametrů do souboru *log.txt*.



Obr. 16: Rozcestník modulů

4.4. Předávání vybraných hodnot

Vybrané vstupní nebo výstupní parametry jsou předávány z modulů do rozcestníku modulů. Může se jednat o nastavení osciloskopu, aktuální hodnotu výstupu generátoru signálu, hodnotu změřenou multimetrem. Tato funkce by měla poskytnout uživateli možnost náhledu na aktuální nastavení modulů bez nutnosti jejich otevření. Předávaná informace není závislá na přihlašovacích právech uživatele a je pro všechny uživatele shodná.

Prvotní verze počítala se spuštěním modulu na pozadí, získání hodnot a ukončením, ale tento postup se ukázal jako zbytečně složitý a neefektivní z pohledu počtu instrukcí. Navíc vzniknul problém s tím, že samotný modul běží až do ukončení v nekonečné smyčce *While* a v případě spouštění na pozadí bylo třeba, aby se smyčka vykonala pouze jednou a nedocházelo tak k zacyklení a zpomalování běhu celého programu. Z těchto důvodů je předávání řešeno pomocí sdílené proměnné. Určitou nevýhodou tohoto řešení může být to, že ke každému modulu je potřeba jedné sdílené proměnné, na druhou stranu informace v této proměnné není využita jednoúčelově, ale je s ní dále pracováno při logování hodnot.

Všechny hodnoty, které jsou určeny pro předání na rozcestník je potřeba převést na textové řetězce. U číselné hodnoty je pro převod nejlepší využít bloku *Number to Fractional String*, který umožňuje pomocí vstupu *Precision* zadat počet desetinných míst, která se mají převést na text. Při ovládání přístrojů je často využíván blok *Menu Ring*. Ten se chová podobně jako *Combo Box* s tím rozdílem, že výstupní hodnotou *Combo Boxu* je textový řetězec. *Menu Ring* má na výstupu celé číslo, které odpovídá indexu vybrané hodnoty z nabídky. V případě multimetru se měřená veličina přepíná pomocí číselné hodnoty na vstupu *Function*. Uživatel v nabídce vybere měření požadované veličiny, ale na výstupu bloku *Menu Ring* je číselná hodnota, která dá konfiguračnímu bloku multimetru informaci o přepnutí na měření požadované veličiny. Převodní tabulku mezi textovou a číselnou hodnotou je možné upravit ve vlastnostech bloku. Pro získání textového řetězce vybrané hodnoty je využít blok *Property Node*. Po převedení všech parametrů na text se jednotlivé řetězce sloučí spolu s vhodnými oddělovači pomocí bloku *Concatenate Strings*, jehož výstupem je složený textový řetězec. Tento řetězec je dále zapsán do sdílené proměnné. V rozcestníku modulů je obsah sdílené proměnné zapsán do textového indikátoru.

4.5. Logování do souboru

Logování slouží ke zpětné kontrole práce s programem. V našem případě je do souboru zapisováno několik informací:

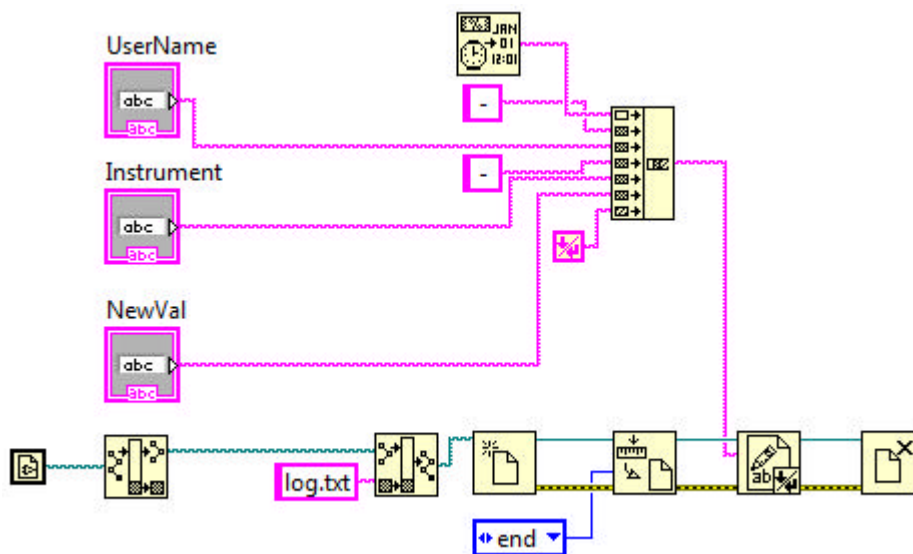
- a) datum a čas změny,
- b) přihlášený uživatel,
- c) identifikátor zařízení,
- d) parametry zařízení.

Ke generování nového záznamu do logovacího souboru dojde v případě změny uživatele nebo změny sledovaných parametrů zařízení.

V návaznosti na logování došlo k malé úpravě sledování předávaných parametrů zařízení. K přepisu hodnoty v indikátoru nedochází okamžitě, ale pouze při změně hodnoty oproti předchozí. LabVIEW obsahuje speciální strukturu pro sledování změn *Event Structure*. Tuto strukturu ale není možné využít ke sledování změn globálních, lokálních ani sdílených proměnných. Ke sledování změny hodnoty v proměnné jsem proto použil zpětnou vazbu *FeedBack Node*. Díky tomu je možné porovnávat aktuální hodnotu v proměnné s hodnotou minulou. V případě, že se tyto dvě hodnoty liší, dojde

ke změně hodnoty v indikátoru a k zavolání podprogramu zajišťujícímu zápis do souboru.

Podprogramu je předáváno uživatelské jméno, identifikátor přístroje a nové parametry přístroje. Pokud je změna vyvolána změnou uživatelského jména, místo identifikátoru a parametrů se předává prázdný řetězec. V podprogramu jsou všechny předané hodnoty, spolu se systémovým časem a oddělovači, sloučeny do jednoho textového řetězce. Na konec řetězce je doplněn znak *End of Line*, který ukončí řádek a posune kurzor na začátek nového řádku. Podprogram otevře soubor *log.txt* v kořenovém adresáři projektu, jen je, stejně jako v případě načítání hesla, potřeba upravit adresu k souboru. Po úspěšném otevření se nalezne konec souboru a zapíše textový řetězec. Posledním krokem je uzavření souboru, po tomto kroku dojde k návratu do hlavního programu.



Obr. 17: Zápis vybraných parametrů do souboru

Základem pro identifikaci připojeného zařízení a aktivaci příslušného modulu je tedy znalost identifikátoru. První výskyt tohoto podprogramu je v rozcestníku modulů. Zde se v pravidelných intervalech testuje pomocí společné části identifikátoru přítomnost připojeného přístroje. Jako návratová hodnota z podprogramu je použita hodnota *Boolean*, na kterou reaguje aktivace nebo deaktivace LED kontrolky a dále tlačítka spuštění modulu na rozcestníku. K aktivaci tlačítka je opět použit *Property Node*.



Obr. 19: Příklad aktivace modulů na rozcestníku

Podprogram se dále vyskytuje v každém modulu, postup vyhledání přístroje je shodný, rozdíl je v použité návratové hodnotě z podprogramu. U modulů se používá výstupní proměnná *Resource Name*, která zde slouží jako náhrada za blok *Instrument Resource Name*. Výstup z podprogramu je tedy přímo přiveden na vstup inicializace přístroje.



Obr. 20: Využití podprogramu při inicializaci přístroje

Tento postup umožňuje připojení přístrojů se stejným typovým označením, ale různými identifikátory bez potřeby ručního výběru přístroje ze seznamu. To proto, že část identifikátoru je vždy shodná a tato shodná část je programem vyhledávána.

Tab. 1: Příklady identifikátorů

Přístroj	Identifikátor	Společná část
Generátor Agilent 33220 A	MY44040254	MY440
	MY44027533	
	MY44040263	
	MY44040257	
Multimetr Agilent 34410 A	MY47009173	MY470
	MY47009193	
	MY47009006	
	MY47009161	

6. Moduly pro měřicí přístroje

Univerzální měřicí rozhraní má sloužit jako nadstavba pro hotové moduly sloužící k ovládání měřicích přístrojů, aktuátorů, laserů, kamer apod. V době vzniku práce bohužel nebyl ani jeden z těchto modulů hotový a funkční a proto jsem se rozhodl, po dohodě s vedoucí bakalářské práce, že naprogramuji moduly pro ovládání měřicích přístrojů v laboratoři A-TK7. V dalších podkapitolách je tedy uveden popis modulů pro generátor signálu, multimetr a laboratorní zdroj. Díky takto široké škále modulů je možné připravit zajímavé úlohy demonstrující vzdálené měření a řízení přístrojů pomocí počítače.

6.1. Generátor signálu Agilent 33220A

Generátor Agilent 33220A je laboratorní generátor signálu umožňující generování sinusového a obdélníkového signálu až do frekvence 20 MHz. Dalšími jeho výstupními signály jsou trojúhelníkový, šumový, pulzní a uživatelsky definovaný tvar signálu. Výstupní D/A převodník má rozlišení 14 bitů a rychlost 50 MS/s. K nadřazenému systému je možné ho připojit a posléze ovládat pomocí komunikačních rozhraní USB, GPIB a LAN.

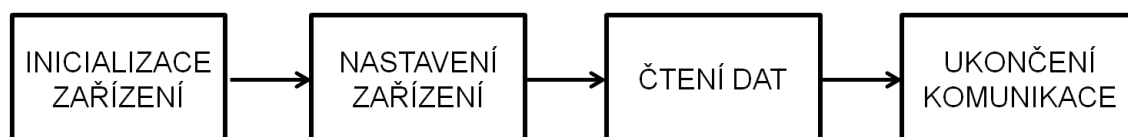
Pro ovládání generátoru pomocí LabVIEW jsem vybral rozhraní USB. Nejprve je nutné stáhnout ze stránek výrobce, popřípadě ze stránek National Instruments, vhodné ovladače. Ovladače pro LabVIEW jsou ve většině případů dodávány ve formě knihovny, kterou je třeba nakopírovat do složky *instr.lib* nacházející se ve složce s nainstalovaným programem (výchozí umístění: "c:\Program Files\National Instruments\LabVIEW 2010\"). Po restartování LabVIEW přibude na paletě *Functions/Instrument IO/Instr Drivers* nová ikona (*AG 33XXX FGen*) zastupující přidané ovladače přístroje. Po jejím otevření jsou nám k dispozici funkční bloky, pomocí kterých probíhá řízení přístroje.

U každého připojeného přístroje platí, že se v první řadě musí nastavit parametry komunikace mezi přístrojem a počítačem. Při použití USB dojde k nastavení parametrů automaticky, přesto se ale musí v programu specifikovat příslušný port, na kterém je zařízení připojeno a dále otevřít komunikační kanál mezi ním a počítačem. Proto každá přístrojová knihovna obsahuje funkční blok s označením *Initialize* (*Init* nebo podobně). Tento blok se stará o navázání komunikace s příslušným zařízením na definovaném portu. V praxi to znamená, že na příslušný port odešle určitou sekvenci znaků a čeká na

odpověď. V případě, že odpověď nepřijde v předepsaném intervalu nebo přijde ve špatném tvaru, funkční blok odešle na výstup odpovídající chybovou hlášku a další vykonávání programu je zastaveno. Pokud je odpověď v pořádku, pokračuje se v programu dále. Každý modul má v první části programu nastavení viditelnosti jednotlivých ovládacích prvků v závislosti na přístupových právech uživatele. Nyní je již možné nastavovat jednotlivé parametry výstupního signálu stejně jako v případě fyzického ovládání generátoru. Modul umožňuje nastavení:

- a) tvaru signálu (sinus, obdélník, rampa, trojúhelník, šum, jednotkový skok),
- b) velikosti amplitudy a její jednotku (špičková nebo efektivní hodnota),
- c) hodnoty frekvence signálu,
- d) offsetu,
- e) velikosti výstupní impedance (50 Ω , vysoká impedance),
- f) střidy signálu (v případě obdélníkového a rampového průběhu).

Po stisknutí tlačítka *START* jsou vybrané hodnoty nastaveny a je sepnut výstup. Ten zůstává sepnut až do další změny nastavení hodnot nebo do vypnutí modulu. Zároveň jsou na Front panelu zobrazeny aktuální parametry signálu, který je přiveden na výstup generátoru. Při vypínání je třeba ukončit komunikaci se zařízením pomocí bloku *Close*, protože pokud by nebyla komunikace ukončena, operační systém počítače si uchová otevřený komunikační kanál s přístrojem a pokud bychom chtěli přistupovat k přístroji pomocí jiného programu, došlo by k chybě a operační systém by hlásil, že připojení k přístroji má vyhrazený jiný program.



Obr. 21: Schéma komunikace se zařízením

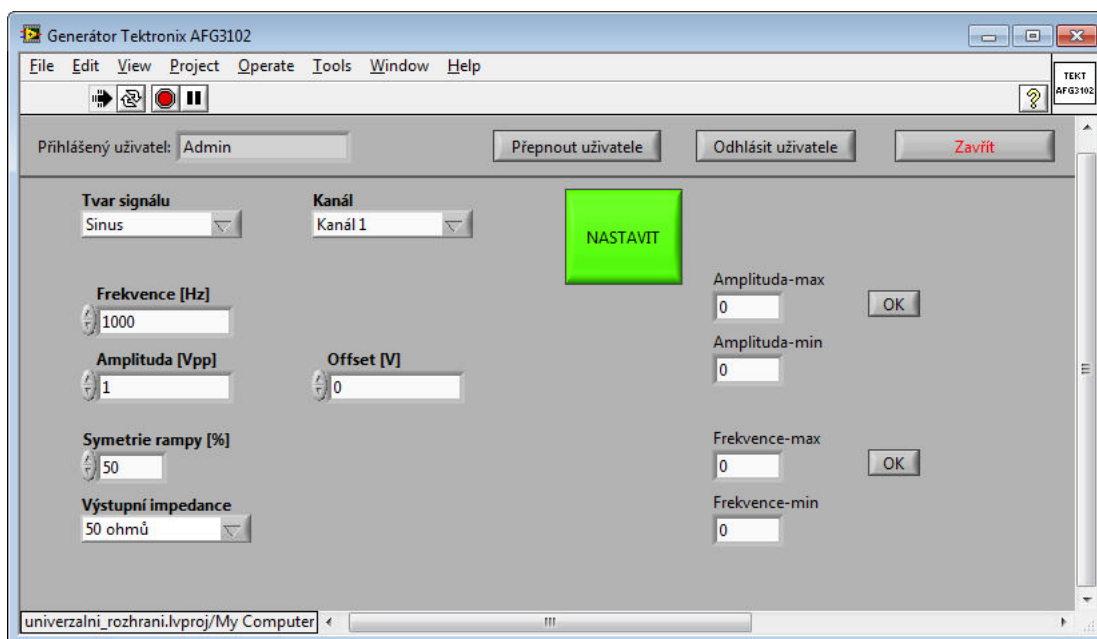
Při stisknutí tlačítka *START* jsou současně sejmuty důležité výstupní hodnoty a ty jsou pomocí podprogramu převedeny na složený textový řetězec, který je zapsán do sdílené proměnné a dále předáván na rozcestník modulů k zobrazení a logování. Při uzavření modulu je hodnota ve sdílené proměnné nahrazena prázdným řetězcem.

6.2. Generátor signálu Tektronix AFG3102

Tektronix AFG3102 je dvoukanálový generátor signálu se sinusovým, obdélníkovým, trojúhelníkovým, rampovým, pulzním a šumovým výstupním signálem. Dále jsou dostupné méně běžné průběhy jako exponenciální růst a pokles, Lorentzův, Gaussův a Haversine průběh. Maximální frekvence sinusového signálu je 100 MHz. D/A převodník na výstupu generátoru je 14 bitový se vzorkovací rychlostí 250 MS/s. Připojení je, stejně jako u generátoru Agilent, možné pomocí USB, GPIB, LAN.

Modul využívá ovládání generátoru pomocí sběrnice USB. Nejdříve je podle přístupových práv nastavena viditelnost ovládacích prvků. Samotné ovládání je velmi podobné jako u generátoru Agilent, bloky mají shodné funkce. Modul na generátoru nastavuje:

- a) tvar signálu,
- b) frekvenci,
- c) amplitudu,
- d) offset,
- e) symetrii rampy,
- f) výstupní impedanci.



Obr. 22: Modul pro generátor signálu Tektronix

U tohoto generátoru má každý blok navíc jednu vstupní hodnotu – kanál, pro který se aktuální hodnota nastavuje. S tím souvisí i úprava zápisu do globální proměnné. Řetězec s výstupními hodnotami není přímo přiveden do globální proměnné, ale je

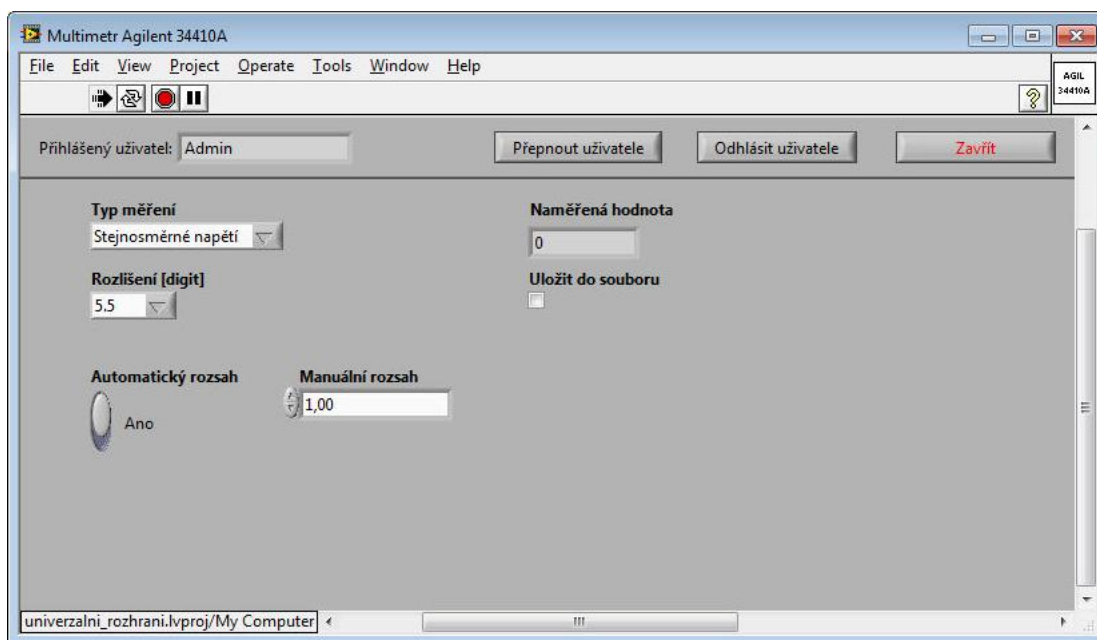
nejprve uložen do lokální proměnné. Pro každý kanál je vyhrazena jedna lokální proměnná. Hodnoty z obou těchto proměnných jsou sloučeny do jednoho řetězce a poté zapsány do sdílené proměnné.

6.3. Multimetr Agilent 34410A

Digitální multimetr Agilent 34410A má 14 měřících funkcí – měření napětí, proudu, odporu, frekvence, kapacity, teploty, atd. Multimetr umožňuje měření s maximální rychlostí 10.000 vzorků/s s přenosem do PC a širokým měřicím rozsahem. Mimo jiné disponuje funkcí automatického měření s vlastní pamětí schopnou uložit až 50.000 vzorků. Komunikace s PC je možná pomocí USB, GPIB, LAN.

Přístroj je k PC připojen přes USB. Ovládání multimetru z těla přístroje je snadné a z této jednoduchosti vychází i ovládání pomocí LabVIEW. Knihovna s ovládacími bloky, která je jako jediná dostupná ihned po instalaci programu, obsahuje podstatně méně bloků, než v předešlých případech. Po navázání komunikace je třeba specifikovat typ měření a poté na výstupu bloku *Read* přečíst naměřenou hodnotu. V ovládacím rozhraní modulu je možné nastavit:

- a) typ měření,
- b) rozlišení,
- c) automatický / manuální rozsah



Obr. 23: Modul pro multimetr Agilent

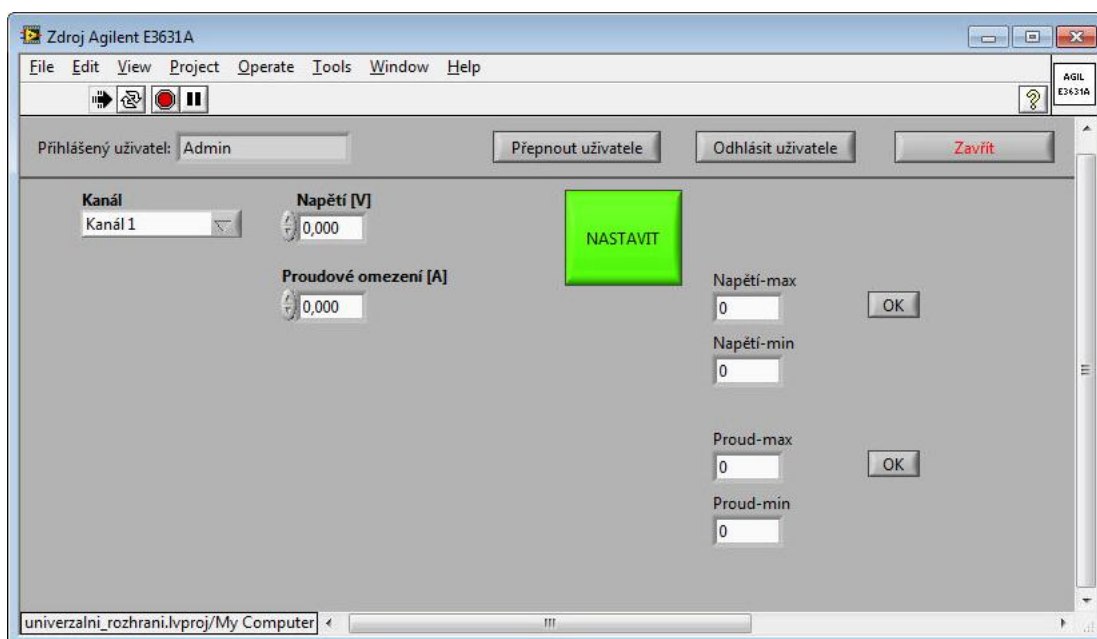
Ve výchozím nastavení se aktuální hodnota načítá desetkrát za sekundu, uživatel může tuto časovou konstantu změnit. Modul navíc umožňuje zápis měřených hodnot do souboru. Povolení zápisu je realizováno zatržením volby *Uložit do souboru*, poté se otevře dialogové okno, ve kterém uživatel vybere soubor pro zápis hodnot a po jeho vybrání začne program pravidelně ukládat naměřené hodnoty.

6.4. Laboratorní zdroj Agilent E3631A

Laboratorní zdroj se třemi nezávislými výstupy $0 - 6\text{ V}$ ($0 - 5\text{ A}$), $0 - \pm 25\text{ V}$ ($0 - 1\text{ A}$) s funkcí spínání těchto výstupů. Připojení k PC je možné pomocí sériové linky RS-232 nebo rozhraní GPIB.

K programování jsem tentokrát využil rozhraní GPIB. Zpočátku se vyskytly problémy při komunikaci, později jsem zjistil, že v počítači je GPIB převodník od firmy Agilent, který ale LabVIEW ve výchozím nastavení ignoruje. Jeho povolení je možné pomocí nástroje *Measurement & Automation Explorer*. V nabídce *Tools/NI-Visa/Visa Options/Passports* je potřeba zatrhnout položku *NiVisaTulip.dll*. Po restartu LabVIEW je možné pracovat i kartami, které nejsou od firmy National Instruments. Pomocí modulu se na přístroji nastavuje:

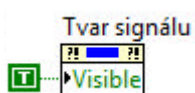
- a) výstupní kanál,
- b) velikost napětí,
- c) proudové omezení.



Obr. 24: Modul pro zdroj Agilent

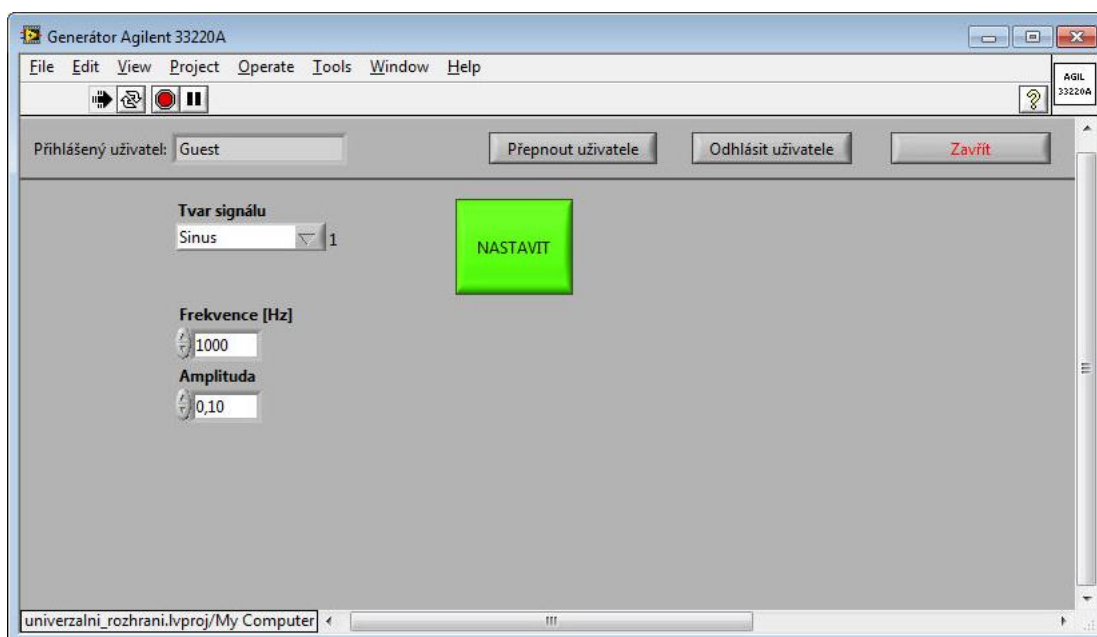
7. Ovládání modulů v závislosti na přístupových právech

Každému prvku na *Front Panelu* může být nastavena viditelnost. Po kliknutí pravým tlačítkem myši na prvek se zobrazí nabídka, ve které vybereme *Create/Property Node/Visible*. Tím se v blokovém diagramu zobrazí funkční blok *Property Node* daného prvku, u kterého chceme zjišťovat nebo měnit viditelnost. Ve výchozím nastavení slouží blok pro čtení viditelnosti prvku s výstupem typu *Boolean* (*True/False*). Po kliknutí pravým tlačítkem je možné změnit blok na zápis pomocí *Change All to Write*. Nyní má blok vstup, opět typu *Boolean*, pomocí kterého je možné nastavovat viditelnost prvku pouhou změnou konstanty na jeho vstupu.

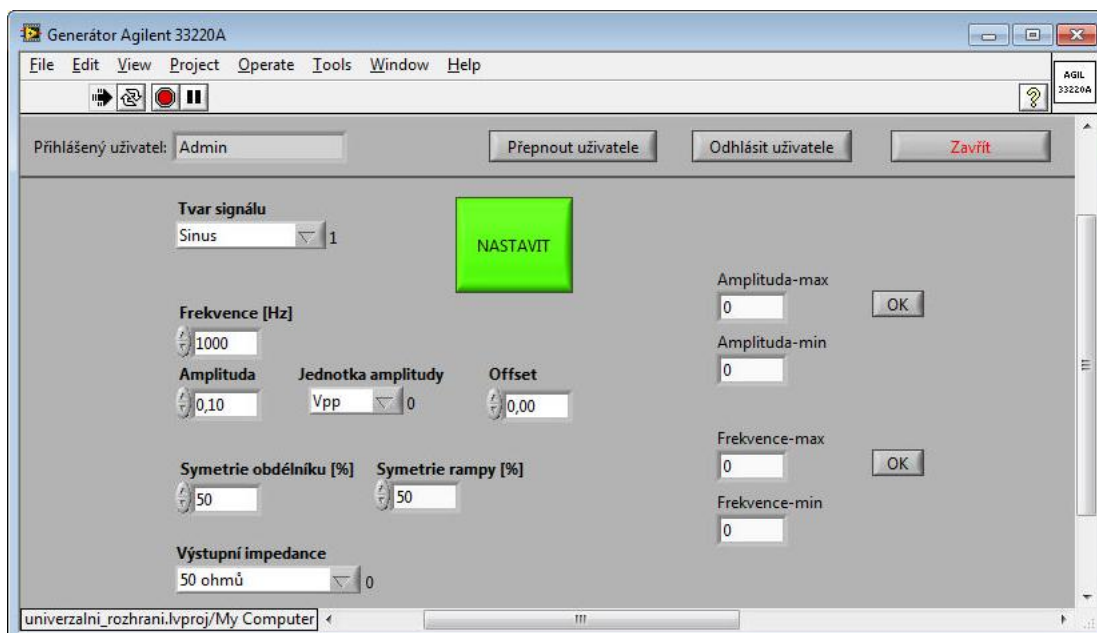


Obr. 25: Nastavení viditelnosti prvku

Tímto způsobem je možné měnit viditelnost všech prvků u všech modulů. V programu je jednoduchá *Case* struktura, na jejíž vstup se přivádí informace o přihlášeném uživateli. *Case* se dělí do třech větví odpovídající jednotlivým uživatelům. V každé této větvi jsou *Property Nody* pro všechny prvky, liší se pouze vstupní konstantou. V současnosti jsou viditelnosti prvků nastavovány přímo ve zdrojovém kódu programu, určitým vylepšením programu do budoucna může být to, že viditelnosti bude možné měnit programově, například z uživatelského účtu Admin.



Obr. 26: Ovládání generátoru Agilent pro uživatele Quest

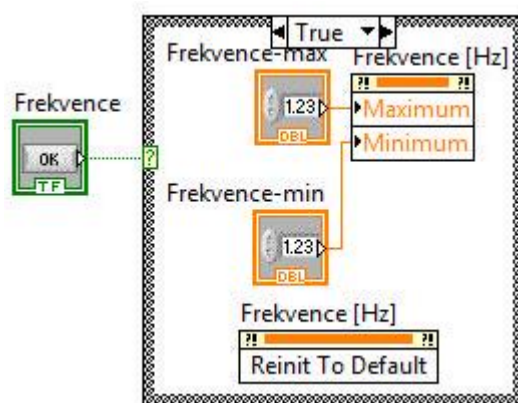


Obr. 27: Ovládání generátoru Agilent pro uživatele Admin

S nastavením uživatelských práv nepřímou souvisí i limitace velikosti výstupních veličin, například u laboratorního zdroje nebo generátoru signálu. Různé úlohy mohou mít různé požadavky na velikosti výstupních veličin, proto není možné omezení výstupu realizovat statickým nastavením hodnot, ale je potřeba, aby bylo programově a za běhu možné měnit krajní hodnoty výstupu. U každého číselného vstupu se dá nastavit maximální a minimální hodnota, kterou lze do konkrétního prvku zapsat. Ve výchozím nastavení jsou maximální a minimální hodnoty dány použitým datovým typem, po zrušení výchozích limitů je nastavení hodnot zcela na volbě uživatele. Samozřejmě platí, že hodnota nemůže být vyšší, než dovoluje daný datový typ. Mimo maximální a minimální hodnoty je možné nastavit velikost přírůstku a způsob, jakým bude prvek reagovat na hodnotu, která je mimo stanovený rozsah. Na výběr jsou dva způsoby reakce:

- a) **Ignore**, který při použití inkrementačního tlačítka nedovolí nastavit vyšší hodnotu než je daný limit, ale při přímém zápisu do prvku hodnotu mimo rozsah ponechá,
- b) **Coerce**, který stejně jako předchozí způsob nedovolí při použití inkrementačního tlačítka nastavit vyšší hodnotu než je daný limit, ale při přímém zápisu hodnotu mimo rozsah změni na nejbližší limitní hodnotu.

Pro účely programu je důležité vybrat v nastavení způsob reakce *Coerce*, tak bude zaručeno, že hodnota nikdy nepřekročí stanovené limity. Veškeré nastavení limitů je dostupné po kliknutí pravým tlačítkem myši na prvek v okně *Properties/Data Entry*. Tímto způsobem nastavení je možné definovat statické limity, ke změně rozsahu v běžícím programu je opět možné využít *Property Node*. V nabídce *Create/Property Node/Data Entry Limits* je možné vybrat maximální hodnotu, minimální hodnotu a velikost přírůstku vstupního prvku. Po výběru vstupního limitu se v blokovém diagramu zobrazí *Property Node* svázaný s daným prvkem. Pro přidání dalšího limitu stačí blok „natáhnout“ směrem dolů. Nakonec je třeba změnit blok na zápis, ve výchozím stavu je určen pro čtení. Posledním krokem je přivedení dvou (tří) číselných vstupů vymezujících vstupní rozsah, případně velikost přírůstku. Výhodou tohoto způsobu nastavení krajních hodnot je to, že jsou pevně svázány s daným vstupním prvkem a hodnoty jsou v něm uloženy. To znamená, že při vypnutí a opětovném spuštění programu zůstanou hodnoty v prvků uloženy. Je vhodné umístit část programu nastavující krajní hodnoty výstupu do smyčky, jejíž vykonání je podmíněné např. stiskem nastavovacího tlačítka a do smyčky dále doplnit nastavení hodnoty do výchozího stavu. Důvodem je, že při zmenšení rozsahu pod aktuální zadanou hodnotu by nemuselo dojít k jejímu zmenšení a tím by krátkodobě byla zadaná hodnota mimo stanovený rozsah. Nastavení do výchozích hodnot se provádí podobně jako změna viditelnosti nebo nastavení rozsahu – pravé tlačítko myši na prvek, který chceme reinitializovat, poté *Create/Invoke Node/Reinicialize to Default*. Blok nemá žádný vstup ani výstup, jeho jedinou funkcí je nastavení hodnot do výchozího stavu.



Obr. 28: Příklad nastavení limitace frekvence

Dále je uvedeno několik tabulek, které názorně shrnují nastavená přístupová práva pro jednotlivé moduly a uživatele.

Tab. 2: Porovnání přístupových práv pro generátor Agilent

uživatel funkce	Quest	User	Admin
<i>tvar signálu</i>	✓	✓	✓
<i>frekvence</i>	✓	✓	✓
<i>amplituda</i>	✓	✓	✓
<i>jednotka amplitudy</i>	X	✓	✓
<i>výstupní impedance</i>	X	✓	✓
<i>střída</i>	X	✓	✓
<i>offset</i>	X	X	✓
<i>limitní hodnoty</i>	X	X	✓

Tab. 3: Porovnání přístupových práv pro generátor Tektronix

uživatel funkce	Quest	User	Admin
<i>kanál</i>	✓	✓	✓
<i>tvar signálu</i>	✓	✓	✓
<i>frekvence</i>	✓	✓	✓
<i>amplituda</i>	✓	✓	✓
<i>výstupní impedance</i>	X	✓	✓
<i>střída</i>	X	✓	✓
<i>offset</i>	X	X	✓
<i>limitní hodnoty</i>	X	X	✓

Tab. 4: Porovnání přístupových práv pro multimetr Agilent

uživatel funkce	Quest	User	Admin
<i>typ měření</i>	✓	✓	✓
<i>rozlišení</i>	X	✓	✓
<i>rozsah</i>	X	X	✓

Tab. 5: Porovnání přístupových práv pro zdroj Agilent

uživatel funkce	Quest	User	Admin
<i>kanál</i>	✓	✓	✓
<i>napětí</i>	✓	✓	✓
<i>proudové omezení</i>	X	✓	✓
<i>limitní hodnoty</i>	X	X	✓

Shrnutí

Práce s prostředím LabVIEW je velmi intuitivní a názorná. Prostředí obsahuje nepřehledné množství funkcí, i z toho důvodu se jedná o jedno s nejrozšířenějších průmyslových prostředí v oblasti sběru dat a zpracování signálu.

Bylo vytvořeno nadstavbové rozhraní k ovládání modulů (programů) v prostředí LabVIEW. Úroveň přístupu k obsahu programu je diferencována pro 3 úrovně – Quest, User a Admin, kde poslední dva jsou chráněny heslem. Heslo je načítáno ze souboru a není tak potřeba kvůli změně hesla zasahovat do zdrojového kódu programu. Mezi účty je možné ve všech částech programu přepínat, změna účtu proběhne okamžitě, spolu se změnou zobrazených ovládacích prvků.

V hlavním okně programu, zajišťujícím zobrazení dostupných modulů, dochází v pravidelných intervalech ke kontrole připojených přístrojů pomocí unikátního označení. Při připojení přístroje dojde během pár vteřin k jeho identifikaci, a pokud je pro něj dostupný modul, dojde k jeho zobrazení na panelu. V hlavním okně jsou dále zobrazeny vybrané parametry ze spuštěných modulů a tyto parametry jsou logovány do souboru.

Jako součásti práce, které nejsou předmětem zadání, vznikly moduly pro ovládání laboratorních měřicích přístrojů – generátoru signálu, multimetru, zdroje. Na těchto modulech byla otestována funkčnost celého programu a možnost připojení přístrojů přes různé sběrnice.

Během tvorby modulů jsem měl v plánu vytvořit ještě modul pro osciloskop. První osciloskop, který jsem si vybral, byl osciloskop Agilent DSO3102A. Pro sběr dat do PC je potřeba dovybavit osciloskop rozšiřujícím komunikačním modulem, který obsahuje rozhraní RS-232 a GPIB. Další osciloskopy, které jsou v laboratoři dostupné, jsou Rigol DS1102D a Rigol DS1102CD. U nich je komunikační rozhraní tvořeno sériovou linkou RS-232 a USB device konektorem pro připojení externího úložiště. Data z těchto osciloskopů se nepodařilo dekodovat, protože výrobce neposkytuje převodovou tabulku. Pro osciloskopy je výrobcem dodáván samostatný program pro čtení dat.

Vzdálený přístup se zdá být nejvýhodnější přes zabudovanou funkci LabVIEW *Remote Panel*, která umožňuje dva typy zobrazení vzdáleného počítače – pomocí LabVIEW nebo pomocí okna webového prohlížeče. Výhody a nevýhody obou přístupů jsou diskutovány v kapitole 3.

Závěr

Naprogramované rozhraní pro prostředí LabVIEW splňuje všechny požadavky, které byly v zadání stanoveny na začátku práce. Vzdálený přístup je realizován pomocí zabudované funkce LabVIEW Remote Panel.

V programu jsou definovány tři uživatelské účty – Quest, User, Admin, přičemž účty User a Admin jsou chráněny heslem. Mezi účty je možné ve všech částech programu přepínat. Změna účtu proběhne okamžitě, spolu se změnou zobrazených ovládacích prvků.

V rámci práce byly navíc vytvořeny moduly pro ovládání přístrojů v laboratoři TK7, jejichž tvorba nebyla zadáním práce. Díky těmto modulům se podařilo celý systém vzdáleného řízení odzkoušet a zdokonalit.

Do budoucna je možné program rozšířit o další moduly, zde není žádné omezení. Může jít o moduly pro další měřicí přístroje, stejně tak i kamery pro záznam a zpracování obrazu, ovládání různých pohonů, senzorů, čidel a jiné. Mimo modulů může být program rozšířen o možnost nastavení uživatelských práv programově a mohou být přidány jazykové mutace programu – s mutací pro anglický jazyk se počítá pro využití v rámci rozšiřující výuky programování v LabVIEW a pro poskytnutí přístupu zahraničním účastníkům.

Seznam použité literatury

- [1] VLACH, Jaroslav; HAVLÍČEK, Josef; VLACH, Martin. *Začínáme s LabVIEW*. 1. vydání, Praha: BEN – technická literatura, 2008. 247 stran. ISBN: 978-80-7300-245-9.
- [2] GOOK, Michael. *Hardwarová rozhraní. Průvodce programátora*. 1. vydání, Brno: Computer Press a.s., 2006. 463 stran. ISBN: 80-251-1019-2.
- [3] OLMR, Vít. *HW server představuje - Sériová linka RS-232*. In: *HW.cz* [online]. 2005 [cit. 2012-02-20]. Dostupné z: <<http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>>
- [4] KAINKA, Burkhard. *USB – měření, řízení a regulace pomocí sběrnice USB*. 1. vydání, Praha: BEN – technická literatura, 2002. 247 stran. ISBN: 80-7300-073-3.
- [5] JAKSCH, Ivan. *Automatizované měřicí systémy se sběrnicí IEEE 488 (GPIB)*. Liberec, 2012. Dostupné z: <http://www.rss.tul.cz/download/cms/03_GPIB_HW.pdf>
- [6] GPIB. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-02-21]. Dostupné z: <<http://cs.wikipedia.org/wiki/GPIB>>
- [7] IEEE-488. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-02-21]. Dostupné z: <<http://en.wikipedia.org/wiki/IEEE-488>>
- [8] NI PCI-6013. In: *NI PCI-6013* [online]. 2012 [cit. 2012-02-28]. Dostupné z: <<http://sine.ni.com/nips/cds/view/p/lang/en/nid/11441>>
- [9] NATIONAL INSTRUMENTS. *LabVIEW User Manual*. Austin, Texas, 2003. Dostupné z: <www.ni.com/pdf/manuals/320999e.pdf>
- [10] Soubor: Cavo-Nullmodem-per-aggiornamento-ricevitori.jpg. [online] [cit. 2012-04-07]. Dostupné z: <<http://edision-nord.it/media/images/org/Cavo-Nullmodem-per-aggiornamento-ricevitori.jpg>>
- [11] Soubor: 1010042348.jpg. [online] [cit. 2012-04-07]. Dostupné z: <<http://content.etilize.com/900/1010042348.jpg>>

- [12] Soubor: 1073261260.600.jpg. [online] [cit. 2012-04-08]. Dostupné z:
<<http://www.cyberresearch.com/images/products/1073261260.600.jpg>>
- [13] Soubor: pci6013_6014_07310205_1.jpg. [online] [cit. 2012-04-10]. Dostupné
z: <[http://sine.ni.com/nips/cds/pages/image?imagepath=/images/products/us/
pci6013_6014_07310205_1.jpg](http://sine.ni.com/nips/cds/pages/image?imagepath=/images/products/us/pci6013_6014_07310205_1.jpg)>

Seznam obrázků

Obr. 1: Okno Block Diagramu s paletou Functions	- 11 -
Obr. 2: Okno Front Panelu s paletou Controls	- 11 -
Obr. 3: Tools paleta	- 12 -
Obr. 4: Připojovací kabel RS-232 s konektorem CANON9 [10].....	- 13 -
Obr. 5: Připojovací USB 2.0 kabel [11]	- 16 -
Obr. 6: Kabel GPIB [12]	- 18 -
Obr. 7: Měřicí karta NI PCI-6013 [13].....	- 20 -
Obr. 8: Připojení ke vzdálené ploše.....	- 22 -
Obr. 9: Připojení ke vzdálenému PC pomocí aplikace TeamViewer.....	- 23 -
Obr. 10: Okno nastavení LabVIEW Web Serveru	- 25 -
Obr. 11: Připojení ke vzdálenému VI souboru pomocí LabVIEW	- 26 -
Obr. 12: Připojení ke vzdálenému souboru pomocí webového prohlížeče	- 27 -
Obr. 13: Blokový diagram programu	- 28 -
Obr. 14: Grafická podoba přihlašovacího okna	- 29 -
Obr. 15: Podprogram pro získání hesla ze souboru.....	- 31 -
Obr. 16: Rozcestník modulů.....	- 32 -
Obr. 17: Zápis vybraných parametrů do souboru.....	- 34 -
Obr. 18: Podprogram pro vyhledání připojených přístrojů	- 35 -
Obr. 19: Příklad aktivace modulů na rozcestníku	- 36 -
Obr. 20: Využití podprogramu při inicializaci přístroje.....	- 36 -
Obr. 21: Schéma komunikace se zařízením	- 38 -
Obr. 22: Modul pro generátor signálu Tektronix	- 39 -
Obr. 23: Modul pro multimetr Agilent.....	- 40 -
Obr. 24: Modul pro zdroj Agilent	- 41 -
Obr. 25: Nastavení viditelnosti prvku	- 42 -

Obr. 26: Ovládání generátoru Agilent pro uživatele Quest.....	- 42 -
Obr. 27: Ovládání generátoru Agilent pro uživatele Admin.....	- 43 -
Obr. 28: Příklad nastavení limitace frekvence	- 44 -

Seznam tabulek

Tab. 1: Příklady identifikátorů	- 36 -
Tab. 2: Porovnání přístupových práv pro generátor Agilent.....	- 45 -
Tab. 3: Porovnání přístupových práv pro generátor Tektronix	- 45 -
Tab. 4: Porovnání přístupových práv pro multimetr Agilent	- 45 -
Tab. 5: Porovnání přístupových práv pro zdroj Agilent.....	- 45 -